# Dataset

The Dataset I am using in my analysis is . This data is available on the link which is https://www.kaggle.com/uciml/indian-liver-patient-records. Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. This dataset was used to evaluate prediction algorithms in an effort to reduce burden on doctors. Each row in the data provides relevant information about the patient. This dataset is used to predict whether a patient is likely to have liver disease based on the input parameters used in the dataset.

liverDisease <- read.csv("Liver\_Disease.csv")

First we will see the actual dimensions of our dataset that how many objects and variables that the dataset have so, for this I will use the R basic function of dimension as follows:

dim(liverDisease)

## [1] 583 11

Here we can see that the total observations are 583 and the total number of attributes are 11.

# Exploration of Dataset

Now, we will see the complete structure of the dataset and also make some transformations in the dataset if needed so.

str(liverDisease)

## 'data.frame': 583 obs. of 11 variables:  
## $ Age : int 65 62 62 58 72 46 26 29 17 55 ...  
## $ Gender : chr "Female" "Male" "Male" "Male" ...  
## $ Total\_Bilirubin : num 0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...  
## $ Direct\_Bilirubin : num 0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...  
## $ Alkaline\_Phosphotase : int 187 699 490 182 195 208 154 202 202 290 ...  
## $ Alamine\_Aminotransferase : int 16 64 60 14 27 19 16 14 22 53 ...  
## $ Aspartate\_Aminotransferase: int 18 100 68 20 59 14 12 11 19 58 ...  
## $ Total\_Protiens : num 6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...  
## $ Albumin : num 3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...  
## $ Albumin\_and\_Globulin\_Ratio: num 0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...  
## $ Dataset : int 1 1 1 1 1 1 1 1 2 1 ...

The following description would help us to the to predict Liver Disease:

: Age of the patient : Gender of the patient : Is a substance produced during the normal breakdown of red blood cells : In the liver, bilirubin is changed into a form that your body can get rid of : is one kind enzyme found in your body, If you have liver disease : is an enzyme that increases in the blood when the liver is damaged : is a blood test that checks for liver damage : test measures the total amount albumin and globulin in your body : is a protein made by your liver : measures the protein ratio in your body : field used to split the data into two sets (patient with liver disease = 2, or no disease = 1)

Now we will see the observations in the dataset as follows:

head(liverDisease)

## Age Gender Total\_Bilirubin Direct\_Bilirubin Alkaline\_Phosphotase  
## 1 65 Female 0.7 0.1 187  
## 2 62 Male 10.9 5.5 699  
## 3 62 Male 7.3 4.1 490  
## 4 58 Male 1.0 0.4 182  
## 5 72 Male 3.9 2.0 195  
## 6 46 Male 1.8 0.7 208  
## Alamine\_Aminotransferase Aspartate\_Aminotransferase Total\_Protiens Albumin  
## 1 16 18 6.8 3.3  
## 2 64 100 7.5 3.2  
## 3 60 68 7.0 3.3  
## 4 14 20 6.8 3.4  
## 5 27 59 7.3 2.4  
## 6 19 14 7.6 4.4  
## Albumin\_and\_Globulin\_Ratio Dataset  
## 1 0.90 1  
## 2 0.74 1  
## 3 0.89 1  
## 4 1.00 1  
## 5 0.40 1  
## 6 1.30 1

We can see that the dataset doesn’t require much transformation because the dataset is almost clean but changes need to be done before processing further.

Here I will change the column name of dataset into Liver\_Disease and will also change the values in the column that patient with liver disease = 1 or no disease = 0 as follows:

names(liverDisease)[names(liverDisease) == "Dataset"] <- "Liver\_Disease"

liverDisease$Liver\_Disease[liverDisease$Liver\_Disease == "1"] <- 0  
  
liverDisease$Liver\_Disease[liverDisease$Liver\_Disease == "2"] <- 1

Now I will also change the Gender columns’ observations into binary form as for Male=1 and for Female=0 as follows:

liverDisease$Gender[liverDisease$Gender == "Female"] <- 0  
  
liverDisease$Gender[liverDisease$Gender == "Male"] <- 1

Here we can see that the column name and the values has been transformed successfully.Now I will make some transformation on the data for the clarity of the data, Here we can see that the Gender and Liver\_Disease variables doesn’t belong to a right data type so I will initially transform them to Factor as follows:

liverDisease$Gender <- factor(liverDisease$Gender)  
  
liverDisease$Liver\_Disease <- as.numeric(liverDisease$Liver\_Disease)  
liverDisease$Liver\_Disease <- factor(liverDisease$Liver\_Disease)

Here I will print the structure of the dataset and also check the if there is any missing values available in the dataset as follows:

str(liverDisease)

## 'data.frame': 583 obs. of 11 variables:  
## $ Age : int 65 62 62 58 72 46 26 29 17 55 ...  
## $ Gender : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 1 1 2 2 ...  
## $ Total\_Bilirubin : num 0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...  
## $ Direct\_Bilirubin : num 0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...  
## $ Alkaline\_Phosphotase : int 187 699 490 182 195 208 154 202 202 290 ...  
## $ Alamine\_Aminotransferase : int 16 64 60 14 27 19 16 14 22 53 ...  
## $ Aspartate\_Aminotransferase: int 18 100 68 20 59 14 12 11 19 58 ...  
## $ Total\_Protiens : num 6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...  
## $ Albumin : num 3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...  
## $ Albumin\_and\_Globulin\_Ratio: num 0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...  
## $ Liver\_Disease : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...

colSums(sapply(liverDisease, is.na))

## Age Gender   
## 0 0   
## Total\_Bilirubin Direct\_Bilirubin   
## 0 0   
## Alkaline\_Phosphotase Alamine\_Aminotransferase   
## 0 0   
## Aspartate\_Aminotransferase Total\_Protiens   
## 0 0   
## Albumin Albumin\_and\_Globulin\_Ratio   
## 0 4   
## Liver\_Disease   
## 0

Here we can see that the Albumin\_and\_Globulin\_Ratio has 4 missing values so here we will decide with the help of percentage data missing, whether we will omit these NAs or change them with the mean or median values as follows:

sum(is.na(liverDisease)) / (nrow(liverDisease) \*ncol(liverDisease))

## [1] 0.000623733

Here the missing data percentage is almost zero so we will omit the NAs from the dataset as follows:

liver <- na.omit(liverDisease)

str(liver)

## 'data.frame': 579 obs. of 11 variables:  
## $ Age : int 65 62 62 58 72 46 26 29 17 55 ...  
## $ Gender : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 1 1 2 2 ...  
## $ Total\_Bilirubin : num 0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...  
## $ Direct\_Bilirubin : num 0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...  
## $ Alkaline\_Phosphotase : int 187 699 490 182 195 208 154 202 202 290 ...  
## $ Alamine\_Aminotransferase : int 16 64 60 14 27 19 16 14 22 53 ...  
## $ Aspartate\_Aminotransferase: int 18 100 68 20 59 14 12 11 19 58 ...  
## $ Total\_Protiens : num 6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...  
## $ Albumin : num 3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...  
## $ Albumin\_and\_Globulin\_Ratio: num 0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...  
## $ Liver\_Disease : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:4] 210 242 254 313  
## ..- attr(\*, "names")= chr [1:4] "210" "242" "254" "313"

After removing the NAs from the dataset so now the total observations are 579 and total attributes are 11.

The first type of analysis we do is Univariate analysis, we will keep going with the preliminary analysis, as to understand that if it is useful to run a Principle Component Analysis and ClusteringAnalysis on the data set.

# Univariate Analysis

## Age

Lets explore the Age variable:

head(liver$Age)

## [1] 65 62 62 58 72 46

length(liver$Age)

## [1] 579

table(liver$Age)

##   
## 4 6 7 8 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31   
## 2 1 2 1 1 1 2 4 2 1 3 5 11 2 3 7 9 3 5 5 14 5 8 7 10 8   
## 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57   
## 20 15 8 11 11 9 21 6 17 5 21 4 3 24 16 6 20 11 23 9 7 6 8 18 4 7   
## 58 60 61 62 63 64 65 66 67 68 69 70 72 73 74 75 78 84 85 90   
## 14 34 5 9 2 6 17 12 1 4 2 9 8 2 4 14 1 1 1 1

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.

unique(liver$Age)

## [1] 65 62 58 72 46 26 29 17 55 57 64 74 61 25 38 33 40 51 63 34 20 84 52 30 48  
## [26] 47 45 42 50 85 35 21 32 31 54 37 66 60 19 75 68 70 49 14 13 18 39 27 36 24  
## [51] 28 53 15 56 44 41 7 22 8 6 4 43 23 12 69 16 78 11 73 67 10 90

length(unique(liver$Age))

## [1] 72

min(liver$Age)

## [1] 4

max(liver$Age)

## [1] 90

Here the total observation of Age is 579 and is a continuous variable that range lies between 4-890. Now, we will see the summary of the Age variable, for further analysis as follows:

summary(liver$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4.00 33.00 45.00 44.78 58.00 90.00

sd(liver$Age)

## [1] 16.22179

var(liver$Age)

## [1] 263.1463

getMode <- function(v) {  
 uniqv <- unique(v)  
 uniqv[which.max(tabulate(match(v, uniqv)))]  
}  
v <- c(liver$Age)  
mode <- getMode(v)  
print(mode)

## [1] 60

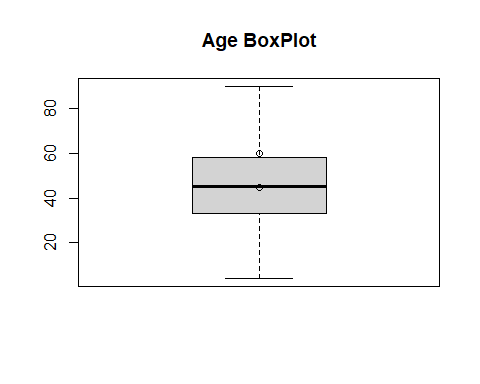
From the above summary we can observe that the Mean (44.78), Median (45.00) and Mode (60) are not equal so the distribution is asymmetrical. Here Mode > Median > Mean so the distribution could be skewed to the left. Now we also confirm this as follows:

library(moments)  
skewness(liver$Age)

## [1] -0.03350366

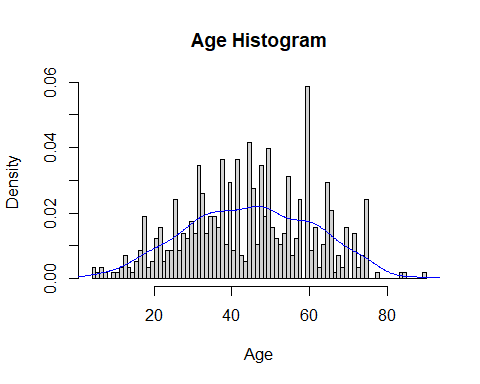
Hence, the skewness is the negative so the distribution is negatively skewed was rightly observed. Also I will plot the Box plot to comparing the distribution of data across dataset as follows:

boxplot(liver$Age, main = "Age BoxPlot")  
points(mean(liver$Age))  
points(mode)



Box plots are graphical displays of the five number summary, So they describe the Minimum (lowest mean relative) lower whisker, Quartile 1 (It separates the lowest 25% observation to the rest of the observations), Median (Which divides the lower 50% observation from the upper 50% observations), Quartile 3 (It divides the upper 25% observation to the rest of the observations) and Maximum. In the above Box plot diagram, there are no outliers and the upper whisker shows the maximum. Now, I will plot a histogram a graphical display of data using bars of different heights to measures distribution of continuous data as follows:

hist(liver$Age, prob = TRUE, breaks = 72, xlab = "Age", main = "Age Histogram")  
lines(density(liver$Age), col='blue')



From the above histogram, we can observe that the diagram has many peaks.Using the density lines, I have approximated the histogram graph. Density provides information about the type of distribution under consideration and allows us to determine whether the attribute is distributed normally or not. From the graph above we can see how the Age variable does not seem to fit to a Normal distribution, there is a peak of the frequency distributions around many values and also a left skewed. We have already observed a negative value of 0.03350366, so we will affirm that the distribution is left skewed. To check whether the distribution is Platykurtic or not, we will check this by following R method as follows:

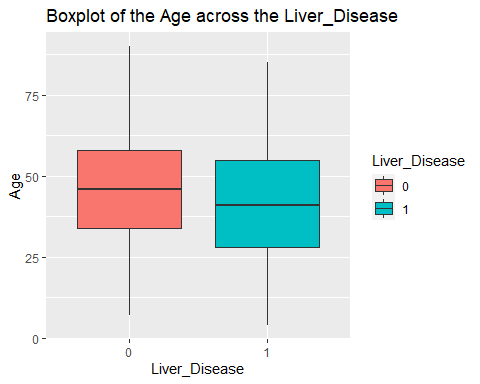
kurtosis(liver$Age)

## [1] 2.429595

The distribution is also Platykurtic, since the value is less than 3.

Now I will plot the Boxplot of the Age variable with the Liver\_Disease variable to check the Liver\_Disease across the Age as follows:

library(ggplot2)  
ggplot(liver, aes(x = Liver\_Disease, y = liver$Age, fill = Liver\_Disease)) +  
 geom\_boxplot() +  
 ylab("Age") +  
 ggtitle("Boxplot of the Age across the Liver\_Disease")



From the graphs we see what the data looks like. The mean age is about 44 which is close to the median of 45. The oldest person in the data is 90 whilst the youngest is 4. The age variable seems to have a bell shaped curve and there seems to be a difference in the ages for the two different liver disease groups. Liver\_Disease = 0 has a higher mean than Liver\_Disease = 1.

### Age Fit for the Data

Now we will try to fit different models to Age distribution evaluating the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC), The lower are the indexes, the better is the fitting. I will evaluate both the single parameter distributions and mixture distributions as follows:

library(MASS)

## Warning: package 'MASS' was built under R version 4.0.4

library(gamlss)

## Warning: package 'gamlss' was built under R version 4.0.5

## Loading required package: splines

## Loading required package: gamlss.data

## Warning: package 'gamlss.data' was built under R version 4.0.4

##   
## Attaching package: 'gamlss.data'

## The following object is masked from 'package:datasets':  
##   
## sleep

## Loading required package: gamlss.dist

## Warning: package 'gamlss.dist' was built under R version 4.0.5

## Loading required package: nlme

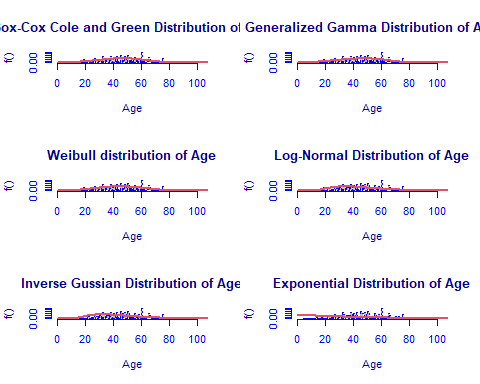
## Loading required package: parallel

## \*\*\*\*\*\*\*\*\*\* GAMLSS Version 5.3-4 \*\*\*\*\*\*\*\*\*\*

## For more on GAMLSS look at https://www.gamlss.com/

## Type gamlssNews() to see new features/changes/bug fixes.

library(splines)  
  
  
par(mfrow=c(3,2))  
  
age.BCCG <- histDist(liver$Age, family=BCCG, nbins = 72, xlab = "Age", main="Box-Cox Cole and Green Distribution of Age")  
  
age.GG <- histDist(liver$Age, family=GG, nbins = 72, xlab = "Age", main="Generalized Gamma Distribution of Age")  
  
age.WEI <- histDist(liver$Age, family=WEI, nbins = 72, xlab = "Age", main="Weibull distribution of Age")  
  
age.LOGNO <- histDist(liver$Age, family=LOGNO, nbins = 72, xlab = "Age", main="Log-Normal Distribution of Age")  
  
age.IG <- histDist(liver$Age, family=IG, nbins=72, xlab = "Age", main = "Inverse Gussian Distribution of Age")  
  
age.EXP<- histDist(liver$Age, family=EXP, nbins=72, xlab = "Age", main = "Exponential Distribution of Age")



library(Matrix)  
library(glmnet)

## Warning: package 'glmnet' was built under R version 4.0.5

## Loaded glmnet 4.1-1

df <- data.frame(Rownames = c("Box-Cox Cole and Green","Generalized Gamma", "Weibull",  
 "Log-Normal", "Inverse Gussian", "Exponential"),  
 AIC = c(AIC(age.BCCG), AIC(age.GG), AIC(age.WEI), AIC(age.LOGNO),  
 AIC(age.IG), AIC(age.EXP)),  
 BIC = c(age.BCCG$sbc, age.GG$sbc, age.WEI$sbc, age.LOGNO$sbc,  
 age.IG$sbc, age.EXP$sbc),  
 df = c(age.BCCG$df.fit, age.GG$df.fit, age.WEI$df.fit,  
 age.LOGNO$df.fit, age.IG$df.fit, age.EXP$df.fit),  
 LogLike = c(logLik(age.BCCG), logLik(age.GG), logLik(age.WEI),  
 logLik(age.LOGNO), logLik(age.IG), logLik(age.EXP)))  
df

## Rownames AIC BIC df LogLike  
## 1 Box-Cox Cole and Green 4870.080 4883.164 3 -2432.040  
## 2 Generalized Gamma 4862.331 4875.415 3 -2428.165  
## 3 Weibull 4867.857 4876.579 2 -2431.928  
## 4 Log-Normal 5021.650 5030.373 2 -2508.825  
## 5 Inverse Gussian 5068.919 5077.642 2 -2532.459  
## 6 Exponential 5562.502 5566.863 1 -2780.251

As we can see, the model with the highest log likelihood (-2428.165) and the lowest AIC (4862.331) and BIC (4875.415) is the Generalized Gamma Distribution. Therefore, based on the greatest likelihood technique, our data fits better in the Generalized Gamma Distribution. Now, I will also compare two models by means of the Likelihood ratio test as we performed above as follows:

library(zoo)

## Warning: package 'zoo' was built under R version 4.0.5

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(lmtest)

## Warning: package 'lmtest' was built under R version 4.0.5

lrtest(age.GG, age.EXP)

## Likelihood ratio test  
##   
## Model 1: gamlssML(formula = liver$Age, family = "GG")  
## Model 2: gamlssML(formula = liver$Age, family = "EXP")  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -2428.2   
## 2 1 -2780.2 -2 704.17 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Under the null hypothesis, we compare the Generalized Gamma distribution to the Exponential distribution under the alternative hypothesis. At any level of significance, we can reject the null hypothesis. Because our distributions fit better in the Generalized Gamma model, we’ll use it.

### Distributions Mixture

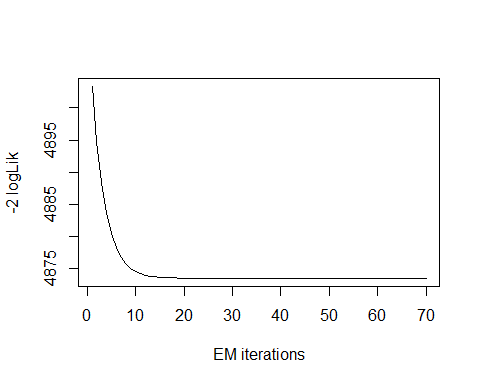
Now, I will also apply the fitting criteria for the distributions with Gamma mixture as follows:

library(gamlss.mx)

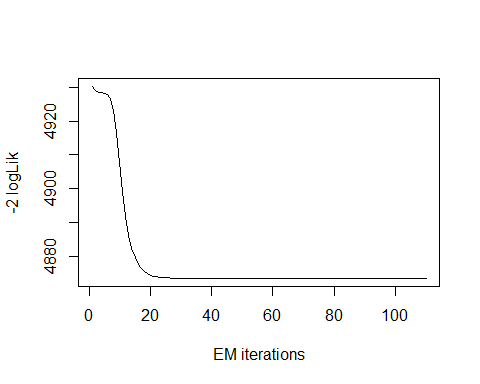
## Warning: package 'gamlss.mx' was built under R version 4.0.5

## Loading required package: nnet

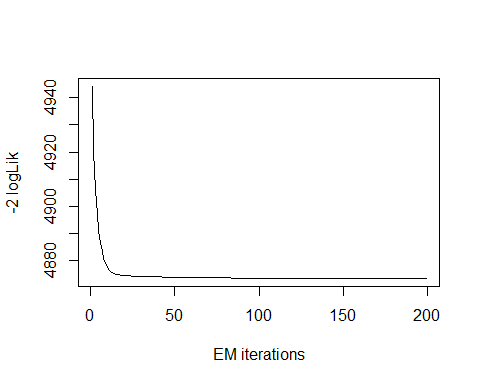
library(nnet)  
mix.gam <- gamlssMXfits(n = 5, liver$Age~1, family = GA, K = 2, data = liver)



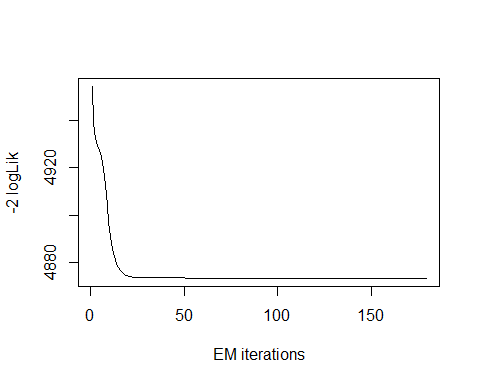
## model= 1



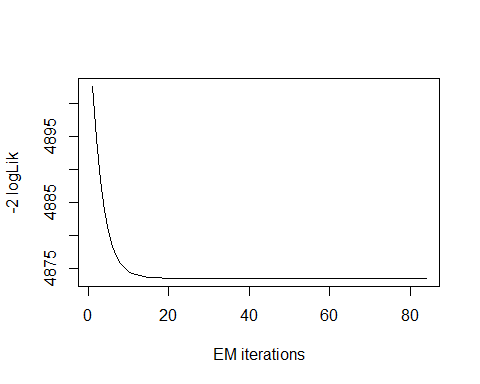
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam)

##   
## Mixing Family: c("GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Age ~ 1, family = GA, K = 2,   
## data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 3.957   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -1.42   
## Mu Coefficients for model: 2   
## (Intercept)   
## 3.516   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -0.7862   
##   
## Estimated probabilities: 0.59742 0.40258   
##   
## Degrees of Freedom for the fit: 5 Residual Deg. of Freedom 574   
## Global Deviance: 4873.44   
## AIC: 4883.44   
## SBC: 4905.25

We can observe that the AIC value of the mixture of Gamma has improved, since it is higher than that of the single Gamma distribution. The current AIC value is 4883.44, whereas the previous value was 4862.331 and the current value of BIC value is 4905.25, whereas the previous value was 4875.415.

logLik(mix.gam)

## 'log Lik.' -2436.722 (df=5)

mix.gam$prob

## [1] 0.59742 0.40258

fitted(mix.gam, "mu")[1]

## [1] 44.78439

fitted(mix.gam, "sigma")[2]

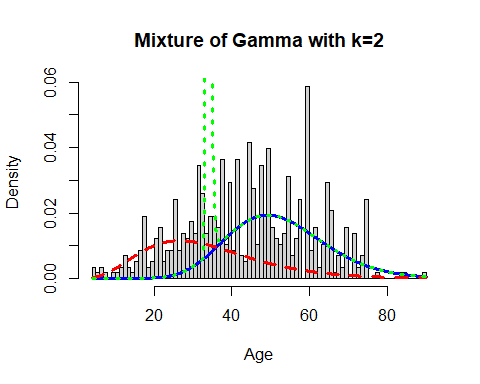
## [1] 44.78439

hist(liver$Age, breaks = 72, xlab = "Age", main="Mixture of Gamma with k=2", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Age, breaks = 72, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Age, breaks = 72, freq = FALSE, plot = FALSE):  
## argument 'freq' is not made use of

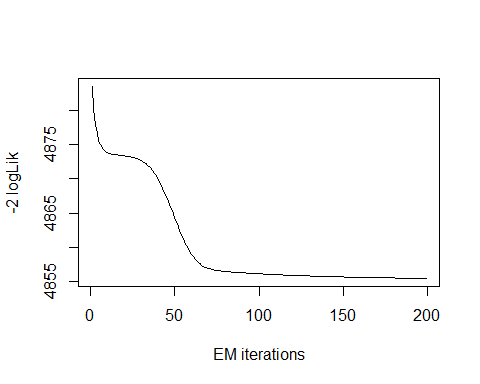
## $breaks  
## [1] 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28  
## [26] 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53  
## [51] 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78  
## [76] 79 80 81 82 83 84 85 86 87 88 89 90  
##   
## $counts  
## [1] 2 1 2 1 0 1 1 2 4 2 1 3 5 11 2 3 7 9 3 5 5 14 5 8 7  
## [26] 10 8 20 15 8 11 11 9 21 6 17 5 21 4 3 24 16 6 20 11 23 9 7 6 8  
## [51] 18 4 7 14 0 34 5 9 2 6 17 12 1 4 2 9 0 8 2 4 14 0 0 1 0  
## [76] 0 0 0 0 1 1 0 0 0 0 1  
##   
## $density  
## [1] 0.003454231 0.001727116 0.003454231 0.001727116 0.000000000 0.001727116  
## [7] 0.001727116 0.003454231 0.006908463 0.003454231 0.001727116 0.005181347  
## [13] 0.008635579 0.018998273 0.003454231 0.005181347 0.012089810 0.015544041  
## [19] 0.005181347 0.008635579 0.008635579 0.024179620 0.008635579 0.013816926  
## [25] 0.012089810 0.017271157 0.013816926 0.034542314 0.025906736 0.013816926  
## [31] 0.018998273 0.018998273 0.015544041 0.036269430 0.010362694 0.029360967  
## [37] 0.008635579 0.036269430 0.006908463 0.005181347 0.041450777 0.027633851  
## [43] 0.010362694 0.034542314 0.018998273 0.039723661 0.015544041 0.012089810  
## [49] 0.010362694 0.013816926 0.031088083 0.006908463 0.012089810 0.024179620  
## [55] 0.000000000 0.058721934 0.008635579 0.015544041 0.003454231 0.010362694  
## [61] 0.029360967 0.020725389 0.001727116 0.006908463 0.003454231 0.015544041  
## [67] 0.000000000 0.013816926 0.003454231 0.006908463 0.024179620 0.000000000  
## [73] 0.000000000 0.001727116 0.000000000 0.000000000 0.000000000 0.000000000  
## [79] 0.000000000 0.001727116 0.001727116 0.000000000 0.000000000 0.000000000  
## [85] 0.000000000 0.001727116  
##   
## $mids  
## [1] 4.5 5.5 6.5 7.5 8.5 9.5 10.5 11.5 12.5 13.5 14.5 15.5 16.5 17.5 18.5  
## [16] 19.5 20.5 21.5 22.5 23.5 24.5 25.5 26.5 27.5 28.5 29.5 30.5 31.5 32.5 33.5  
## [31] 34.5 35.5 36.5 37.5 38.5 39.5 40.5 41.5 42.5 43.5 44.5 45.5 46.5 47.5 48.5  
## [46] 49.5 50.5 51.5 52.5 53.5 54.5 55.5 56.5 57.5 58.5 59.5 60.5 61.5 62.5 63.5  
## [61] 64.5 65.5 66.5 67.5 68.5 69.5 70.5 71.5 72.5 73.5 74.5 75.5 76.5 77.5 78.5  
## [76] 79.5 80.5 81.5 82.5 83.5 84.5 85.5 86.5 87.5 88.5 89.5  
##   
## $xname  
## [1] "liver$Age"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Age), max(liver$Age), length = length(liver$Age)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Age), max(liver$Age),  
 length = length(liver$Age)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Age), max(liver$Age), length = length(liver$Age)),  
 mix.gam[["prob"]][2]\*dGA(seq(min(liver$Age), max(liver$Age),  
 length = length(liver$Age)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Age), max(liver$Age), length = length(liver$Age)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Age), max(liver$Age),  
 length = length(liver$Age)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam[["prob"]][2]\*dRG(seq(min(liver$Age), max(liver$Age),  
 length = length(liver$Age)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")

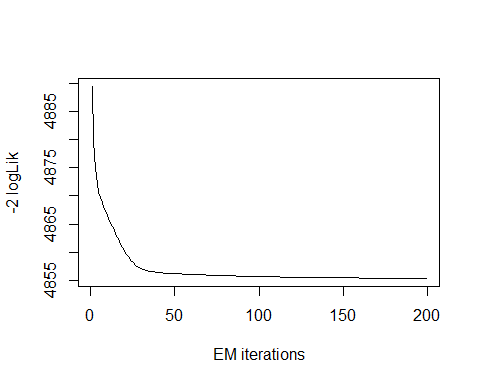


Now, I will again apply the fitting criteria for the distributions with Gamma mixture with the k = 3 as follows:

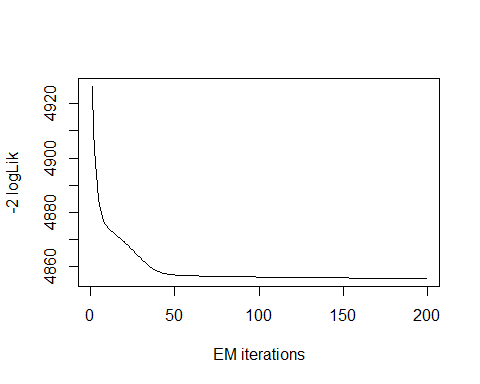
mix.gam.3 <- gamlssMXfits(n = 5, liver$Age~1, family = GA, K = 3, data = liver)



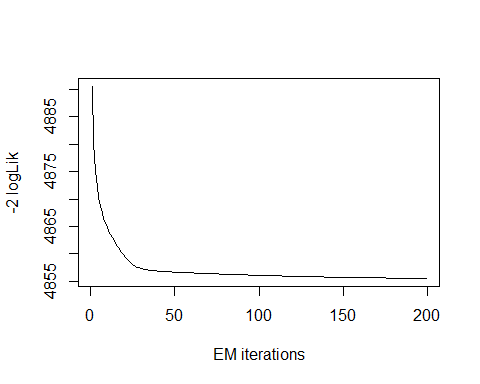
## model= 1



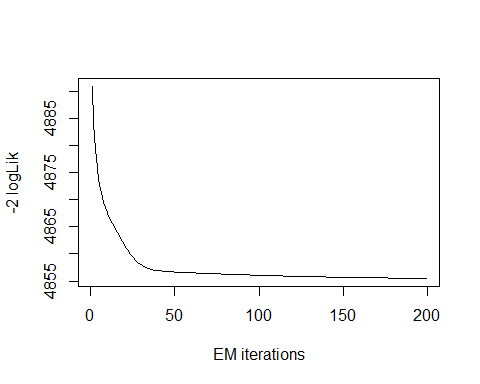
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam.3)

##   
## Mixing Family: c("GA", "GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Age ~ 1, family = GA, K = 3,   
## data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 4.135   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -2.036   
## Mu Coefficients for model: 2   
## (Intercept)   
## 3.773   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -1.297   
## Mu Coefficients for model: 3   
## (Intercept)   
## 3.35   
## Sigma Coefficients for model: 3   
## (Intercept)   
## -0.6822   
##   
## Estimated probabilities: 0.2297243 0.5638963 0.2063795   
##   
## Degrees of Freedom for the fit: 8 Residual Deg. of Freedom 571   
## Global Deviance: 4855.36   
## AIC: 4871.36   
## SBC: 4906.25

logLik(mix.gam.3)

## 'log Lik.' -2427.682 (df=8)

mix.gam.3$prob

## [1] 0.2297243 0.5638963 0.2063795

fitted(mix.gam.3, "mu")[1]

## [1] 44.77739

fitted(mix.gam.3, "sigma")[2]

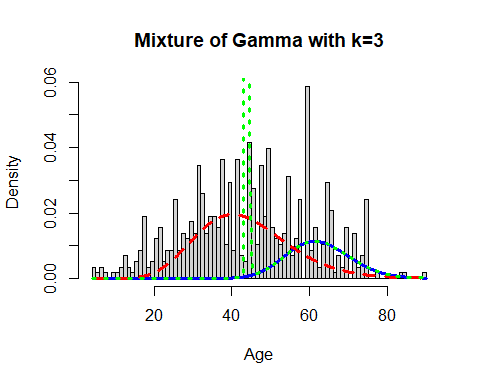
## [1] 44.77739

hist(liver$Age, breaks = 72, xlab = "Age", main="Mixture of Gamma with k=3", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam.3[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam.3[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam.3[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam.3[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Age, breaks = 72, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Age, breaks = 72, freq = FALSE, plot = FALSE):  
## argument 'freq' is not made use of

## $breaks  
## [1] 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28  
## [26] 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53  
## [51] 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78  
## [76] 79 80 81 82 83 84 85 86 87 88 89 90  
##   
## $counts  
## [1] 2 1 2 1 0 1 1 2 4 2 1 3 5 11 2 3 7 9 3 5 5 14 5 8 7  
## [26] 10 8 20 15 8 11 11 9 21 6 17 5 21 4 3 24 16 6 20 11 23 9 7 6 8  
## [51] 18 4 7 14 0 34 5 9 2 6 17 12 1 4 2 9 0 8 2 4 14 0 0 1 0  
## [76] 0 0 0 0 1 1 0 0 0 0 1  
##   
## $density  
## [1] 0.003454231 0.001727116 0.003454231 0.001727116 0.000000000 0.001727116  
## [7] 0.001727116 0.003454231 0.006908463 0.003454231 0.001727116 0.005181347  
## [13] 0.008635579 0.018998273 0.003454231 0.005181347 0.012089810 0.015544041  
## [19] 0.005181347 0.008635579 0.008635579 0.024179620 0.008635579 0.013816926  
## [25] 0.012089810 0.017271157 0.013816926 0.034542314 0.025906736 0.013816926  
## [31] 0.018998273 0.018998273 0.015544041 0.036269430 0.010362694 0.029360967  
## [37] 0.008635579 0.036269430 0.006908463 0.005181347 0.041450777 0.027633851  
## [43] 0.010362694 0.034542314 0.018998273 0.039723661 0.015544041 0.012089810  
## [49] 0.010362694 0.013816926 0.031088083 0.006908463 0.012089810 0.024179620  
## [55] 0.000000000 0.058721934 0.008635579 0.015544041 0.003454231 0.010362694  
## [61] 0.029360967 0.020725389 0.001727116 0.006908463 0.003454231 0.015544041  
## [67] 0.000000000 0.013816926 0.003454231 0.006908463 0.024179620 0.000000000  
## [73] 0.000000000 0.001727116 0.000000000 0.000000000 0.000000000 0.000000000  
## [79] 0.000000000 0.001727116 0.001727116 0.000000000 0.000000000 0.000000000  
## [85] 0.000000000 0.001727116  
##   
## $mids  
## [1] 4.5 5.5 6.5 7.5 8.5 9.5 10.5 11.5 12.5 13.5 14.5 15.5 16.5 17.5 18.5  
## [16] 19.5 20.5 21.5 22.5 23.5 24.5 25.5 26.5 27.5 28.5 29.5 30.5 31.5 32.5 33.5  
## [31] 34.5 35.5 36.5 37.5 38.5 39.5 40.5 41.5 42.5 43.5 44.5 45.5 46.5 47.5 48.5  
## [46] 49.5 50.5 51.5 52.5 53.5 54.5 55.5 56.5 57.5 58.5 59.5 60.5 61.5 62.5 63.5  
## [61] 64.5 65.5 66.5 67.5 68.5 69.5 70.5 71.5 72.5 73.5 74.5 75.5 76.5 77.5 78.5  
## [76] 79.5 80.5 81.5 82.5 83.5 84.5 85.5 86.5 87.5 88.5 89.5  
##   
## $xname  
## [1] "liver$Age"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Age), max(liver$Age), length = length(liver$Age)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Age), max(liver$Age),  
 length = length(liver$Age)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Age), max(liver$Age), length = length(liver$Age)),  
 mix.gam.3[["prob"]][2]\*dGA(seq(min(liver$Age), max(liver$Age),  
 length = length(liver$Age)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Age), max(liver$Age), length = length(liver$Age)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Age), max(liver$Age),  
 length = length(liver$Age)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam.3[["prob"]][2]\*dRG(seq(min(liver$Age), max(liver$Age),  
 length = length(liver$Age)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")



mix.gm.tb <- data.frame(row.names=c('Gamma Mixture, K=3','Gamma Mixture, K=2', "Generalized Gamma"),  
 AIC=c(mix.gam.3$aic, mix.gam$aic, AIC(age.GG)),  
 BIC=c(mix.gam.3$sbc, mix.gam$sbc, age.GG$sbc))  
mix.gm.tb

## AIC BIC  
## Gamma Mixture, K=3 4871.363 4906.254  
## Gamma Mixture, K=2 4883.443 4905.250  
## Generalized Gamma 4862.331 4875.415

We can observe that the AIC value of the mixture of Gamma with k=3 and k=2 has increased, since values are higher than that of the single Gamma distribution. The previous AIC value is 4862, whereas the current value which is higher is 4871 and the previous BIC value was 4875, whereas the current value which is higher is 4906.

## Gender

Lets analyze the features of Gender variable as follows:

length(liver$Gender)

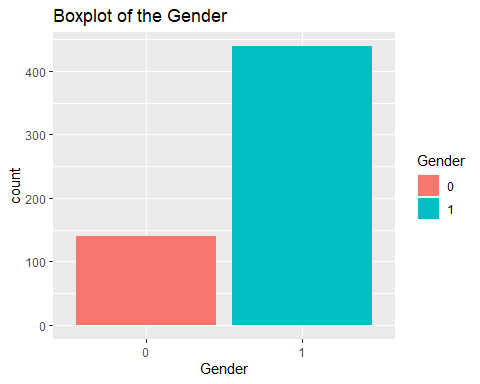
## [1] 579

table(liver$Gender)

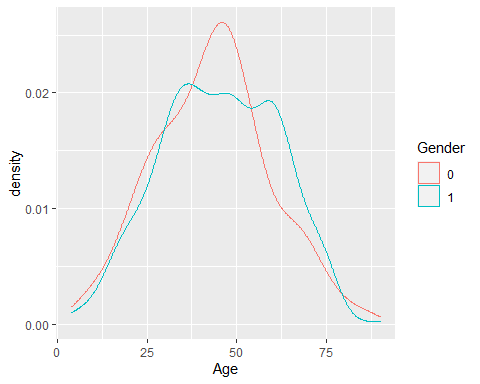
##   
## 0 1   
## 140 439

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.Here we can see the Gender variable is a categorical variable that can take two values 0 for Female and 1 for Male.

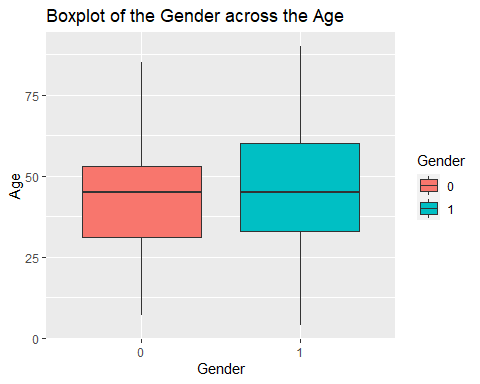
library(ggplot2)  
ggplot(liver, aes(x = Gender, fill = Gender)) +   
geom\_bar() +  
 ggtitle("Boxplot of the Gender")



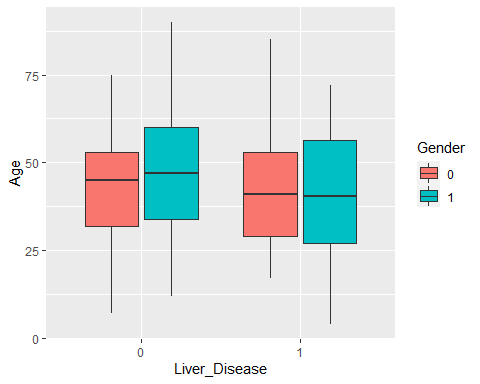
ggplot(liver, aes(x=Age, color=Gender)) +   
 geom\_density()



liver$Gender[liver$Gender == "Female"] <- 0  
liver$Gender[liver$Gender == "Male"] <- 1  
  
  
ggplot(liver, aes(x = Gender, y = Age, fill = Gender)) +  
 geom\_boxplot() +  
 ylab("Age") +  
 ggtitle("Boxplot of the Gender across the Age")



ggplot(liver, aes(Liver\_Disease, Age)) +   
 geom\_boxplot(aes(fill = Gender))



There are more males than females in the entire dataset. The mean age is higher for males than females in the data, both genders have a similar density shape when you look at the age. There are a few graphical ways of looking at this just to make sure we are doing the work right. When looking at the breakdown of the Age, Gender and Liver\_Disease we see some interesting things. When we look at the females, the mean age is higher for the Liver\_Disease = 0 and the same is true for Liver\_Disease = 1. Within the Liver\_Disease = 1 group the mean ages are pretty close but in the Liver\_Disease = 0 ,the males have a higher mean age. We can also seen that the barplot shows our data set contains data belonging to 140 Females and 439 Males. So from both plot and table() we can see that our data is not balanced and the study includes more Males than Females.

## Total Bilirubin

Lets explore the Total\_Bilirubin variable:

head(liver$Total\_Bilirubin)

## [1] 0.7 10.9 7.3 1.0 3.9 1.8

length(liver$Total\_Bilirubin)

## [1] 579

table(liver$Total\_Bilirubin)

##   
## 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9   
## 1 5 45 77 90 56 28 19 8 11 13 5 8 11 14 8   
## 2 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3 3.1 3.2 3.3 3.4 3.5   
## 8 4 8 4 5 2 5 9 4 6 2 2 3 3 1 3   
## 3.6 3.7 3.8 3.9 4 4.1 4.2 4.4 4.5 4.7 4.9 5 5.2 5.3 5.5 5.7   
## 2 3 1 4 3 2 2 1 2 1 1 2 1 2 1 1   
## 5.8 5.9 6.2 6.3 6.6 6.7 6.8 7.1 7.3 7.4 7.5 7.7 7.9 8 8.2 8.6   
## 5 1 1 1 1 2 4 2 3 1 1 1 1 1 1 1   
## 8.7 8.9 9.4 10.2 10.6 10.9 11 11.1 11.3 11.5 12.1 12.7 14.1 14.2 14.5 14.8   
## 1 3 1 1 1 2 1 1 1 1 1 2 1 1 1 1   
## 15 15.2 15.6 15.8 15.9 16.4 16.6 16.7 17.3 17.7 18 18.4 18.5 19.6 19.8 20   
## 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1   
## 20.2 22.5 22.6 22.7 22.8 23 23.2 23.3 25 26.3 27.2 27.7 30.5 30.8 32.6 42.8   
## 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1   
## 75   
## 1

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.

unique(liver$Total\_Bilirubin)

## [1] 0.7 10.9 7.3 1.0 3.9 1.8 0.9 0.6 2.7 1.1 1.6 2.2 2.9 6.8 1.9  
## [16] 4.1 6.2 4.0 2.6 1.3 14.2 1.4 2.4 18.4 3.1 8.9 0.8 2.8 2.0 5.7  
## [31] 8.6 5.8 5.2 3.8 6.6 0.5 5.3 3.2 1.2 12.7 15.9 18.0 23.0 22.7 1.7  
## [46] 3.0 11.3 4.7 4.2 3.5 5.9 8.7 11.0 11.5 4.5 75.0 22.8 14.1 14.8 10.6  
## [61] 8.0 1.5 2.1 6.3 2.3 27.2 2.5 3.6 30.5 16.4 14.5 18.5 23.2 3.7 3.3  
## [76] 7.1 6.7 22.6 7.5 5.0 4.9 8.2 0.4 7.4 23.3 7.9 3.4 19.8 32.6 17.7  
## [91] 20.0 26.3 4.4 9.4 30.8 19.6 15.8 5.5 20.2 27.7 11.1 10.2 42.8 15.2 16.6  
## [106] 17.3 22.5 16.7 7.7 15.6 12.1 25.0 15.0

length(unique(liver$Total\_Bilirubin))

## [1] 113

min(liver$Total\_Bilirubin)

## [1] 0.4

max(liver$Total\_Bilirubin)

## [1] 75

Here the total observation of Age is 579 and is a continuous variable that range lies between 0.4-75. Now, we will see the summary of the Total\_Bilirubin variable, for further analysis as follows:

summary(liver$Total\_Bilirubin)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.400 0.800 1.000 3.315 2.600 75.000

sd(liver$Total\_Bilirubin)

## [1] 6.227716

var(liver$Total\_Bilirubin)

## [1] 38.78445

getMode <- function(v) {  
 uniqv <- unique(v)  
 uniqv[which.max(tabulate(match(v, uniqv)))]  
}  
v <- c(liver$Total\_Bilirubin)  
mode <- getMode(v)  
print(mode)

## [1] 0.8

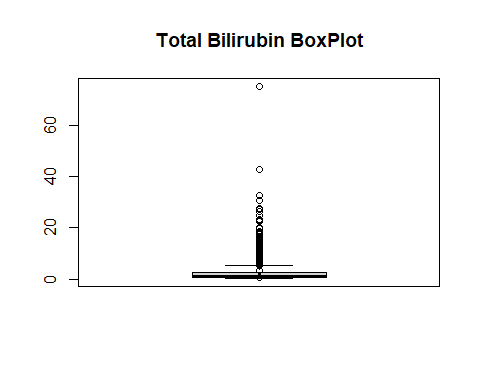
From the above summary we can observe that the Mean (3.315), Median (1.000) and Mode (0.8) are not equal so the distribution is asymmetrical. Here Mean > Median > Mode so the distribution could be positive and skewed to the right. Now we also confirm this as follows:

library(moments)  
skewness(liver$Total\_Bilirubin)

## [1] 4.878088

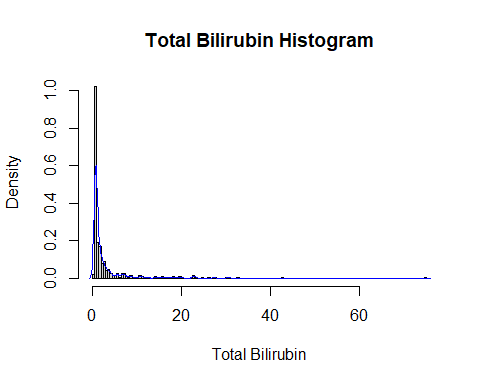
Hence, the skewness is the negative so the distribution is positively skewed was rightly observed. Also I will plot the Box plot to comparing the distribution of data across dataset as follows:

boxplot(liver$Total\_Bilirubin, main = "Total Bilirubin BoxPlot")  
points(mean(liver$Total\_Bilirubin))  
points(mode)



Box plots are graphical displays of the five number summary, So they describe the Minimum (lowest mean relative) lower whisker, Quartile 1 (It separates the lowest 25% observation to the rest of the observations), Median (Which divides the lower 50% observation from the upper 50% observations), Quartile 3 (It divides the upper 25% observation to the rest of the observations) and Maximum. In the above Box plot diagram, there are many outliers and the upper whisker shows the highest mean relative. Now, I will plot a histogram a graphical display of data using bars of different heights to measures distribution of continuous data as follows:

hist(liver$Total\_Bilirubin, prob = TRUE, breaks = 113, xlab = "Total Bilirubin", main = "Total Bilirubin Histogram")  
lines(density(liver$Total\_Bilirubin), col='blue')



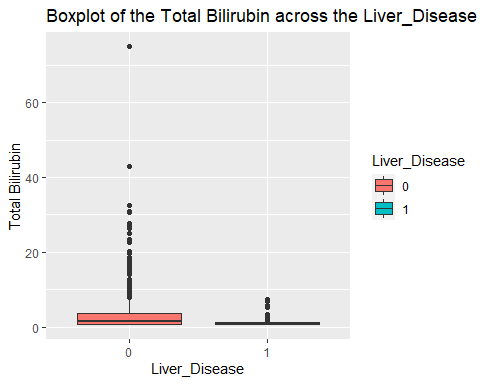
From the above histogram, we can observe that the diagram has many peaks.Using the density lines, I have approximated the histogram graph. Density provides information about the type of distribution under consideration and allows us to determine whether the attribute is distributed normally or not. From the graph above we can see how the Total\_Bilirubin variable does not seem to fit to a Normal distribution, there is a peak of the frequency distributions around many values and also a right skewed. We have already observed a positive value of 4.878088, so we will affirm that the distribution is right skewed. To check whether the distribution is Platykurtic or not, we will check this by following R method as follows:

kurtosis(liver$Total\_Bilirubin)

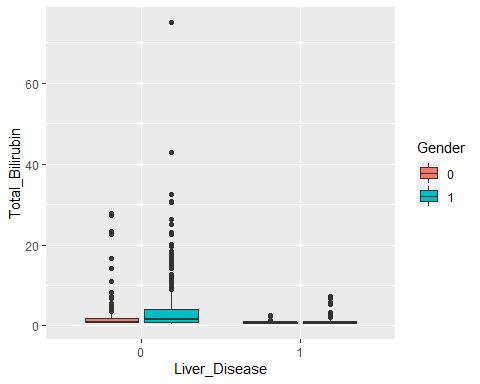
## [1] 39.59283

The distribution is also leptokurtic, since the value is greater than 3.

ggplot(liver, aes(x = Liver\_Disease, y = Total\_Bilirubin, fill = Liver\_Disease)) +  
 geom\_boxplot() +  
 ylab("Total Bilirubin") +  
 ggtitle("Boxplot of the Total Bilirubin across the Liver\_Disease")



ggplot(liver, aes(Liver\_Disease, Total\_Bilirubin)) +   
 geom\_boxplot(aes(fill = Gender))

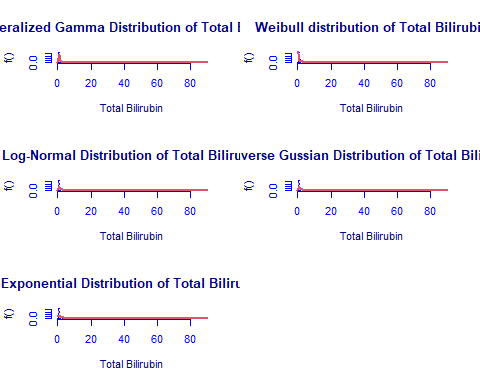


There is a wide range for the Total Bilirubin. The data is skewed, very skewed, we have a maximum value of 75 with the mean of 3.299, median of 1. There is a difference between the mean Total Bilirubin for the Liver\_Disease and this also seems to be true when you break it down further by gender.

### Total Bilirubin Fit for the Data

Now we will try to fit different models to Total\_Bilirubin distribution evaluating the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC), The lower are the indexes, the better is the fitting. I will evaluate both the single parameter distributions and mixture distributions as follows:

library(MASS)  
library(gamlss)  
  
par(mfrow=c(3,2))  
  
tb.GG <- histDist(liver$Total\_Bilirubin, family=GG, nbins = 113, xlab = "Total Bilirubin", main="Generalized Gamma Distribution of Total Bilirubin")  
  
tb.WEI <- histDist(liver$Total\_Bilirubin, family=WEI, nbins = 113, xlab = "Total Bilirubin", main="Weibull distribution of Total Bilirubin")  
  
tb.LOGNO <- histDist(liver$Total\_Bilirubin, family=LOGNO, nbins = 113, xlab = "Total Bilirubin", main="Log-Normal Distribution of Total Bilirubin")  
  
tb.IG <- histDist(liver$Total\_Bilirubin, family=IG, nbins=113, xlab = "Total Bilirubin", main = "Inverse Gussian Distribution of Total Bilirubin")  
  
tb.EXP<- histDist(liver$Total\_Bilirubin, family=EXP, nbins=113, xlab = "Total Bilirubin", main = "Exponential Distribution of Total Bilirubin")



library(Matrix)  
library(glmnet)  
df <- data.frame(Rownames = c("Generalized Gamma", "Weibull", "Log-Normal",  
 "Inverse Gussian", "Exponential"),  
 AIC = c(AIC(tb.GG), AIC(tb.WEI), AIC(tb.LOGNO), AIC(tb.IG),  
 AIC(tb.EXP)),  
 BIC = c(tb.GG$sbc, tb.WEI$sbc, tb.LOGNO$sbc, tb.IG$sbc,  
 tb.EXP$sbc),  
 df = c(tb.GG$df.fit, tb.WEI$df.fit, tb.LOGNO$df.fit,  
 tb.IG$df.fit, tb.EXP$df.fit),  
 LogLike = c(logLik(tb.GG), logLik(tb.WEI), logLik(tb.LOGNO),  
 logLik(tb.IG), logLik(tb.EXP)))  
df

## Rownames AIC BIC df LogLike  
## 1 Generalized Gamma 1833.703 1846.787 3 -913.8516  
## 2 Weibull 2481.176 2489.898 2 -1238.5878  
## 3 Log-Normal 2211.160 2219.883 2 -1103.5801  
## 4 Inverse Gussian 2143.455 2152.178 2 -1069.7275  
## 5 Exponential 2547.944 2552.305 1 -1272.9718

As we can see, the model with the highest log likelihood (-913.8516) and the lowest AIC (1833.703) and BIC (1846.787) is the Generalized Gamma Distribution. Therefore, based on the greatest likelihood technique, our data fits better in the Generalized Gamma Distribution. Now, I will also compare two models by means of the Likelihood ratio test as we performed above as follows:

library(zoo)  
library(lmtest)  
lrtest(tb.GG, tb.EXP)

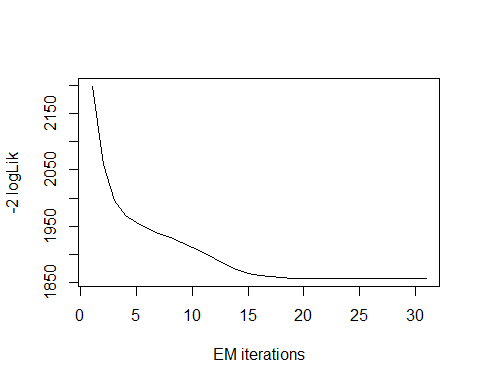
## Likelihood ratio test  
##   
## Model 1: gamlssML(formula = liver$Total\_Bilirubin, family = "GG")  
## Model 2: gamlssML(formula = liver$Total\_Bilirubin, family = "EXP")  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -913.85   
## 2 1 -1272.97 -2 718.24 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Under the null hypothesis, we compare the Generalized Gamma distribution to the Exponential distribution under the alternative hypothesis. At any level of significance, we can reject the null hypothesis. Because our distributions fit better in the Generalized Gamma model, we’ll use it.

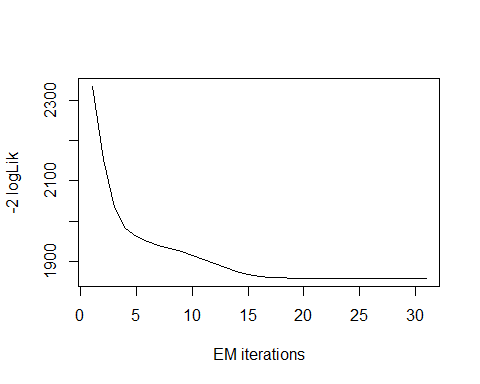
### Distributions Mixture

Now, I will also apply the fitting criteria for the distributions with Gamma mixture as follows:

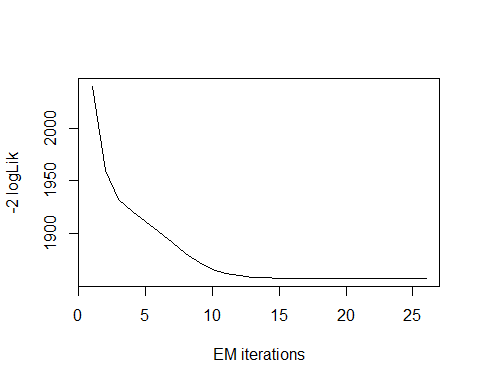
library(gamlss.mx)  
mix.gam <- gamlssMXfits(n = 5, liver$Total\_Bilirubin~1, family = GA, K = 2, data = liver)



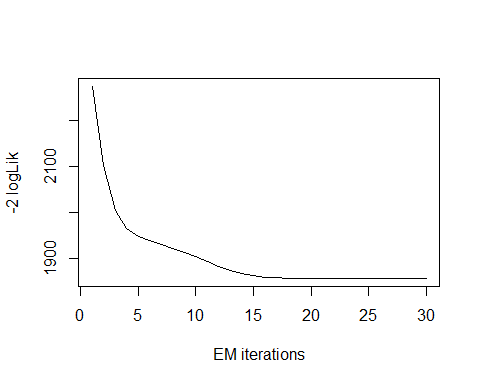
## model= 1



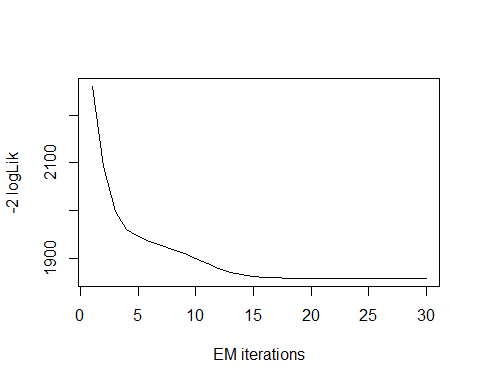
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam)

##   
## Mixing Family: c("GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Total\_Bilirubin ~ 1, family = GA,   
## K = 2, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 1.795   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -0.005117   
## Mu Coefficients for model: 2   
## (Intercept)   
## -0.2179   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -1.693   
##   
## Estimated probabilities: 0.4813882 0.5186118   
##   
## Degrees of Freedom for the fit: 5 Residual Deg. of Freedom 574   
## Global Deviance: 1857.52   
## AIC: 1867.52   
## SBC: 1889.32

We can observe that the AIC value of the mixture of Gamma has not improved, since it is higher than that of the single Gamma distribution. The current AIC value is 1867.52, whereas the previous value was 1833.703 and the current BIC value is 1889.32, whereas the previous value was 1846.787.

logLik(mix.gam)

## 'log Lik.' -928.7576 (df=5)

mix.gam$prob

## [1] 0.4813882 0.5186118

fitted(mix.gam, "mu")[1]

## [1] 3.315516

fitted(mix.gam, "sigma")[2]

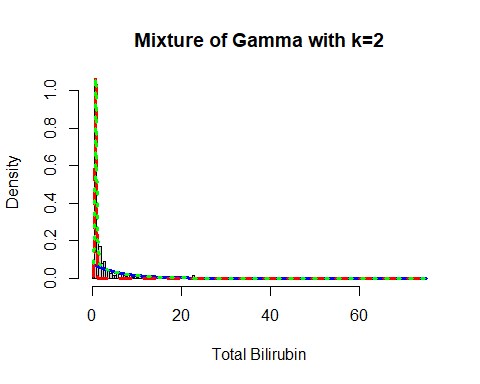
## [1] 3.315516

hist(liver$Total\_Bilirubin, breaks = 113, xlab = "Total Bilirubin", main="Mixture of Gamma with k=2", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Total\_Bilirubin, breaks = 113, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Total\_Bilirubin, breaks = 113, freq = FALSE, :  
## argument 'freq' is not made use of

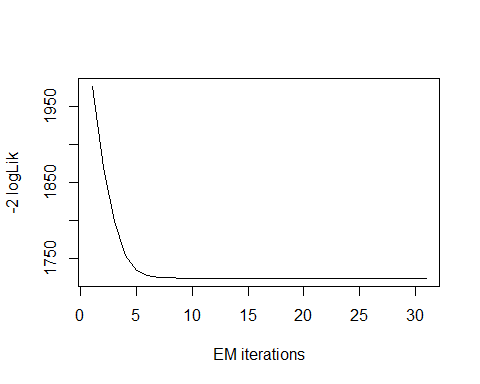
## $breaks  
## [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0  
## [16] 7.5 8.0 8.5 9.0 9.5 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0 14.5  
## [31] 15.0 15.5 16.0 16.5 17.0 17.5 18.0 18.5 19.0 19.5 20.0 20.5 21.0 21.5 22.0  
## [46] 22.5 23.0 23.5 24.0 24.5 25.0 25.5 26.0 26.5 27.0 27.5 28.0 28.5 29.0 29.5  
## [61] 30.0 30.5 31.0 31.5 32.0 32.5 33.0 33.5 34.0 34.5 35.0 35.5 36.0 36.5 37.0  
## [76] 37.5 38.0 38.5 39.0 39.5 40.0 40.5 41.0 41.5 42.0 42.5 43.0 43.5 44.0 44.5  
## [91] 45.0 45.5 46.0 46.5 47.0 47.5 48.0 48.5 49.0 49.5 50.0 50.5 51.0 51.5 52.0  
## [106] 52.5 53.0 53.5 54.0 54.5 55.0 55.5 56.0 56.5 57.0 57.5 58.0 58.5 59.0 59.5  
## [121] 60.0 60.5 61.0 61.5 62.0 62.5 63.0 63.5 64.0 64.5 65.0 65.5 66.0 66.5 67.0  
## [136] 67.5 68.0 68.5 69.0 69.5 70.0 70.5 71.0 71.5 72.0 72.5 73.0 73.5 74.0 74.5  
## [151] 75.0  
##   
## $counts  
## [1] 6 296 56 49 23 26 12 13 7 4 4 7 2 7 7 3 1 5  
## [19] 1 0 1 4 3 0 1 2 0 0 3 2 1 3 1 2 1 2  
## [37] 3 0 0 3 1 0 0 0 1 4 2 0 0 1 0 0 1 0  
## [55] 1 1 0 0 0 0 2 1 0 0 0 1 0 0 0 0 0 0  
## [73] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0  
## [91] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [109] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [127] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [145] 0 0 0 0 0 1  
##   
## $density  
## [1] 0.020725389 1.022452504 0.193436960 0.169257340 0.079447323 0.089810017  
## [7] 0.041450777 0.044905009 0.024179620 0.013816926 0.013816926 0.024179620  
## [13] 0.006908463 0.024179620 0.024179620 0.010362694 0.003454231 0.017271157  
## [19] 0.003454231 0.000000000 0.003454231 0.013816926 0.010362694 0.000000000  
## [25] 0.003454231 0.006908463 0.000000000 0.000000000 0.010362694 0.006908463  
## [31] 0.003454231 0.010362694 0.003454231 0.006908463 0.003454231 0.006908463  
## [37] 0.010362694 0.000000000 0.000000000 0.010362694 0.003454231 0.000000000  
## [43] 0.000000000 0.000000000 0.003454231 0.013816926 0.006908463 0.000000000  
## [49] 0.000000000 0.003454231 0.000000000 0.000000000 0.003454231 0.000000000  
## [55] 0.003454231 0.003454231 0.000000000 0.000000000 0.000000000 0.000000000  
## [61] 0.006908463 0.003454231 0.000000000 0.000000000 0.000000000 0.003454231  
## [67] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [73] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [79] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [85] 0.000000000 0.003454231 0.000000000 0.000000000 0.000000000 0.000000000  
## [91] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [97] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [103] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [109] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [115] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [121] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [127] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [133] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [139] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [145] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.003454231  
##   
## $mids  
## [1] 0.25 0.75 1.25 1.75 2.25 2.75 3.25 3.75 4.25 4.75 5.25 5.75  
## [13] 6.25 6.75 7.25 7.75 8.25 8.75 9.25 9.75 10.25 10.75 11.25 11.75  
## [25] 12.25 12.75 13.25 13.75 14.25 14.75 15.25 15.75 16.25 16.75 17.25 17.75  
## [37] 18.25 18.75 19.25 19.75 20.25 20.75 21.25 21.75 22.25 22.75 23.25 23.75  
## [49] 24.25 24.75 25.25 25.75 26.25 26.75 27.25 27.75 28.25 28.75 29.25 29.75  
## [61] 30.25 30.75 31.25 31.75 32.25 32.75 33.25 33.75 34.25 34.75 35.25 35.75  
## [73] 36.25 36.75 37.25 37.75 38.25 38.75 39.25 39.75 40.25 40.75 41.25 41.75  
## [85] 42.25 42.75 43.25 43.75 44.25 44.75 45.25 45.75 46.25 46.75 47.25 47.75  
## [97] 48.25 48.75 49.25 49.75 50.25 50.75 51.25 51.75 52.25 52.75 53.25 53.75  
## [109] 54.25 54.75 55.25 55.75 56.25 56.75 57.25 57.75 58.25 58.75 59.25 59.75  
## [121] 60.25 60.75 61.25 61.75 62.25 62.75 63.25 63.75 64.25 64.75 65.25 65.75  
## [133] 66.25 66.75 67.25 67.75 68.25 68.75 69.25 69.75 70.25 70.75 71.25 71.75  
## [145] 72.25 72.75 73.25 73.75 74.25 74.75  
##   
## $xname  
## [1] "liver$Total\_Bilirubin"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Total\_Bilirubin), max(liver$Total\_Bilirubin),  
 length = length(liver$Total\_Bilirubin)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Total\_Bilirubin),  
 max(liver$Total\_Bilirubin), length = length(liver$Total\_Bilirubin)),  
 mu = mu.hat1, sigma = sigma.hat1), lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Total\_Bilirubin), max(liver$Total\_Bilirubin),  
 length = length(liver$Total\_Bilirubin)),  
 mix.gam[["prob"]][2]\*dGA(seq(min(liver$Total\_Bilirubin),  
 max(liver$Total\_Bilirubin), length = length(liver$Total\_Bilirubin)),  
 mu = mu.hat2, sigma = sigma.hat2), lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Total\_Bilirubin), max(liver$Total\_Bilirubin),  
 length = length(liver$Total\_Bilirubin)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Total\_Bilirubin),  
 max(liver$Total\_Bilirubin), length = length(liver$Total\_Bilirubin)),  
 mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam[["prob"]][2]\*dRG(seq(min(liver$Total\_Bilirubin),  
 max(liver$Total\_Bilirubin), length = length(liver$Total\_Bilirubin)),  
 mu = mu.hat2, sigma = sigma.hat2), lty = 3, lwd = 3, col = "green")

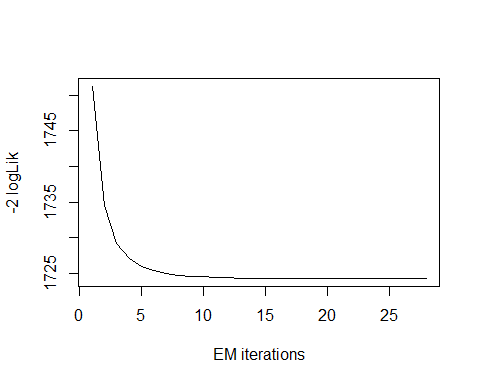


Now, I will again apply the fitting criteria for the distributions with Gamma mixture with the k = 3 as follows:

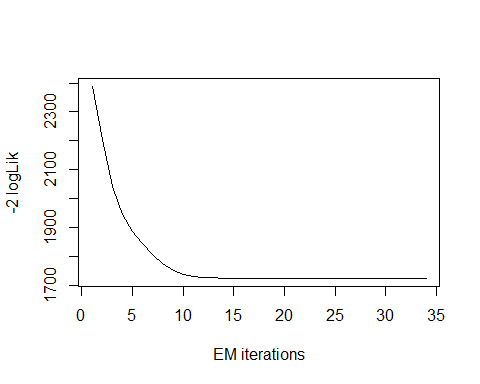
mix.gam.3 <- gamlssMXfits(n = 5, liver$Total\_Bilirubin~1, family = GA, K = 3,  
 data = liver)



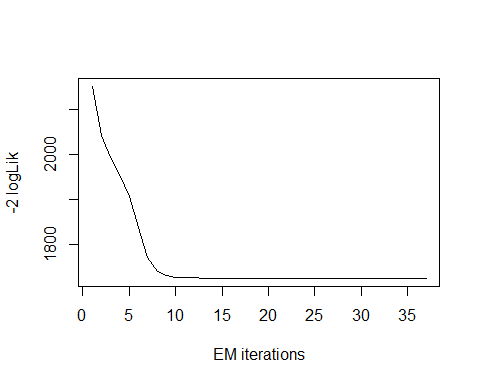
## model= 1



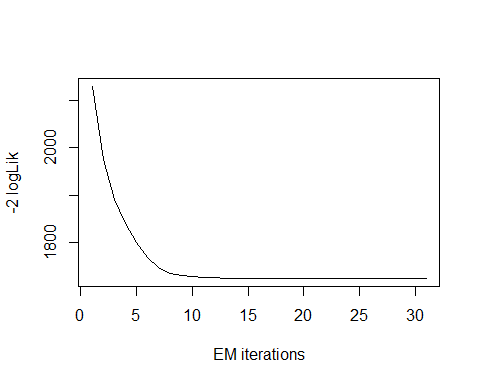
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam.3)

##   
## Mixing Family: c("GA", "GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Total\_Bilirubin ~ 1, family = GA,   
## K = 3, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## -0.2409   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -1.765   
## Mu Coefficients for model: 2   
## (Intercept)   
## 0.6743   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -0.7985   
## Mu Coefficients for model: 3   
## (Intercept)   
## 2.433   
## Sigma Coefficients for model: 3   
## (Intercept)   
## -0.2131   
##   
## Estimated probabilities: 0.4978339 0.2966577 0.2055084   
##   
## Degrees of Freedom for the fit: 8 Residual Deg. of Freedom 571   
## Global Deviance: 1724.3   
## AIC: 1740.3   
## SBC: 1775.19

logLik(mix.gam.3)

## 'log Lik.' -862.1487 (df=8)

mix.gam.3$prob

## [1] 0.4978339 0.2966577 0.2055084

fitted(mix.gam.3, "mu")[1]

## [1] 3.315623

fitted(mix.gam.3, "sigma")[2]

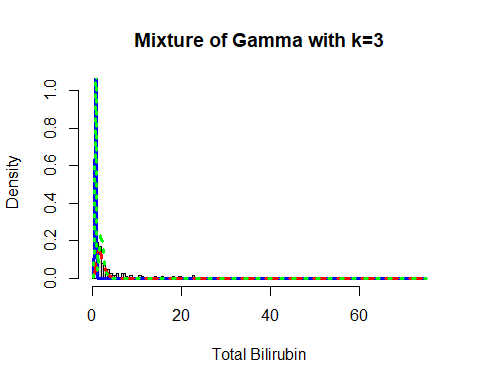
## [1] 3.315623

hist(liver$Total\_Bilirubin, breaks = 113, xlab = "Total Bilirubin", main="Mixture of Gamma with k=3", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam.3[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam.3[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam.3[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam.3[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Total\_Bilirubin, breaks = 113, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Total\_Bilirubin, breaks = 113, freq = FALSE, :  
## argument 'freq' is not made use of

## $breaks  
## [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0  
## [16] 7.5 8.0 8.5 9.0 9.5 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0 14.5  
## [31] 15.0 15.5 16.0 16.5 17.0 17.5 18.0 18.5 19.0 19.5 20.0 20.5 21.0 21.5 22.0  
## [46] 22.5 23.0 23.5 24.0 24.5 25.0 25.5 26.0 26.5 27.0 27.5 28.0 28.5 29.0 29.5  
## [61] 30.0 30.5 31.0 31.5 32.0 32.5 33.0 33.5 34.0 34.5 35.0 35.5 36.0 36.5 37.0  
## [76] 37.5 38.0 38.5 39.0 39.5 40.0 40.5 41.0 41.5 42.0 42.5 43.0 43.5 44.0 44.5  
## [91] 45.0 45.5 46.0 46.5 47.0 47.5 48.0 48.5 49.0 49.5 50.0 50.5 51.0 51.5 52.0  
## [106] 52.5 53.0 53.5 54.0 54.5 55.0 55.5 56.0 56.5 57.0 57.5 58.0 58.5 59.0 59.5  
## [121] 60.0 60.5 61.0 61.5 62.0 62.5 63.0 63.5 64.0 64.5 65.0 65.5 66.0 66.5 67.0  
## [136] 67.5 68.0 68.5 69.0 69.5 70.0 70.5 71.0 71.5 72.0 72.5 73.0 73.5 74.0 74.5  
## [151] 75.0  
##   
## $counts  
## [1] 6 296 56 49 23 26 12 13 7 4 4 7 2 7 7 3 1 5  
## [19] 1 0 1 4 3 0 1 2 0 0 3 2 1 3 1 2 1 2  
## [37] 3 0 0 3 1 0 0 0 1 4 2 0 0 1 0 0 1 0  
## [55] 1 1 0 0 0 0 2 1 0 0 0 1 0 0 0 0 0 0  
## [73] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0  
## [91] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [109] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [127] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [145] 0 0 0 0 0 1  
##   
## $density  
## [1] 0.020725389 1.022452504 0.193436960 0.169257340 0.079447323 0.089810017  
## [7] 0.041450777 0.044905009 0.024179620 0.013816926 0.013816926 0.024179620  
## [13] 0.006908463 0.024179620 0.024179620 0.010362694 0.003454231 0.017271157  
## [19] 0.003454231 0.000000000 0.003454231 0.013816926 0.010362694 0.000000000  
## [25] 0.003454231 0.006908463 0.000000000 0.000000000 0.010362694 0.006908463  
## [31] 0.003454231 0.010362694 0.003454231 0.006908463 0.003454231 0.006908463  
## [37] 0.010362694 0.000000000 0.000000000 0.010362694 0.003454231 0.000000000  
## [43] 0.000000000 0.000000000 0.003454231 0.013816926 0.006908463 0.000000000  
## [49] 0.000000000 0.003454231 0.000000000 0.000000000 0.003454231 0.000000000  
## [55] 0.003454231 0.003454231 0.000000000 0.000000000 0.000000000 0.000000000  
## [61] 0.006908463 0.003454231 0.000000000 0.000000000 0.000000000 0.003454231  
## [67] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [73] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [79] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [85] 0.000000000 0.003454231 0.000000000 0.000000000 0.000000000 0.000000000  
## [91] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [97] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [103] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [109] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [115] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [121] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [127] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [133] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [139] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [145] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.003454231  
##   
## $mids  
## [1] 0.25 0.75 1.25 1.75 2.25 2.75 3.25 3.75 4.25 4.75 5.25 5.75  
## [13] 6.25 6.75 7.25 7.75 8.25 8.75 9.25 9.75 10.25 10.75 11.25 11.75  
## [25] 12.25 12.75 13.25 13.75 14.25 14.75 15.25 15.75 16.25 16.75 17.25 17.75  
## [37] 18.25 18.75 19.25 19.75 20.25 20.75 21.25 21.75 22.25 22.75 23.25 23.75  
## [49] 24.25 24.75 25.25 25.75 26.25 26.75 27.25 27.75 28.25 28.75 29.25 29.75  
## [61] 30.25 30.75 31.25 31.75 32.25 32.75 33.25 33.75 34.25 34.75 35.25 35.75  
## [73] 36.25 36.75 37.25 37.75 38.25 38.75 39.25 39.75 40.25 40.75 41.25 41.75  
## [85] 42.25 42.75 43.25 43.75 44.25 44.75 45.25 45.75 46.25 46.75 47.25 47.75  
## [97] 48.25 48.75 49.25 49.75 50.25 50.75 51.25 51.75 52.25 52.75 53.25 53.75  
## [109] 54.25 54.75 55.25 55.75 56.25 56.75 57.25 57.75 58.25 58.75 59.25 59.75  
## [121] 60.25 60.75 61.25 61.75 62.25 62.75 63.25 63.75 64.25 64.75 65.25 65.75  
## [133] 66.25 66.75 67.25 67.75 68.25 68.75 69.25 69.75 70.25 70.75 71.25 71.75  
## [145] 72.25 72.75 73.25 73.75 74.25 74.75  
##   
## $xname  
## [1] "liver$Total\_Bilirubin"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Total\_Bilirubin), max(liver$Total\_Bilirubin),  
 length = length(liver$Total\_Bilirubin)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Total\_Bilirubin),  
 max(liver$Total\_Bilirubin), length = length(liver$Total\_Bilirubin)),  
 mu = mu.hat1, sigma = sigma.hat1), lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Total\_Bilirubin), max(liver$Total\_Bilirubin),  
 length = length(liver$Total\_Bilirubin)),  
 mix.gam.3[["prob"]][2]\*dGA(seq(min(liver$Total\_Bilirubin),  
 max(liver$Total\_Bilirubin), length = length(liver$Total\_Bilirubin)),  
 mu = mu.hat2, sigma = sigma.hat2), lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Total\_Bilirubin), max(liver$Total\_Bilirubin),  
 length = length(liver$Total\_Bilirubin)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Total\_Bilirubin),  
 max(liver$Total\_Bilirubin), length = length(liver$Total\_Bilirubin)),  
 mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam.3[["prob"]][2]\*dRG(seq(min(liver$Total\_Bilirubin),  
 max(liver$Total\_Bilirubin), length = length(liver$Total\_Bilirubin)),  
 mu = mu.hat2, sigma = sigma.hat2), lty = 3, lwd = 3, col = "green")



mix.gm.tb <- data.frame(row.names=c('Gamma Mixture, K=3','Gamma Mixture, K=2', "Generalized Gamma"),  
 AIC=c(mix.gam.3$aic, mix.gam$aic, AIC(tb.GG)),  
 BIC=c(mix.gam.3$sbc, mix.gam$sbc, tb.GG$sbc))  
mix.gm.tb

## AIC BIC  
## Gamma Mixture, K=3 1740.297 1775.188  
## Gamma Mixture, K=2 1867.515 1889.322  
## Generalized Gamma 1833.703 1846.787

We can observe that the AIC value of the mixture of Gamma with k=3 has increased, since k=2 and single Gamma distribution values are higher than that of the current AIC and BIC values. The current AIC value is 1740.297, whereas the previous value which is higher is 1833.703 and the current BIC value is 1775.188, whereas the current value which is higher is 1846.787. Here we can clearly see that our data fits better in the Gamma Mixture Distribution with k=3.

## Direct Bilirubin

Lets explore the Direct Bilirubin variable:

head(liver$Direct\_Bilirubin)

## [1] 0.1 5.5 4.1 0.4 2.0 0.7

length(liver$Direct\_Bilirubin)

## [1] 579

table(liver$Direct\_Bilirubin)

##   
## 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4 1.5 1.6   
## 63 192 50 21 20 15 11 22 7 13 7 10 12 7 6 11   
## 1.7 1.8 1.9 2 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3 3.2 3.3   
## 2 3 2 3 4 2 5 1 3 1 3 1 1 5 6 1   
## 3.6 3.7 3.9 4 4.1 4.2 4.3 4.5 4.6 4.9 5 5.1 5.2 5.5 5.6 6   
## 4 2 1 3 2 1 2 2 1 1 2 1 1 1 1 1   
## 6.1 6.2 6.4 7 7.2 7.6 7.7 7.8 8.2 8.4 8.5 8.8 8.9 9 9.5 10   
## 1 1 1 2 1 2 1 1 2 2 2 2 1 2 3 1   
## 10.2 10.4 10.8 11.3 11.4 11.7 11.8 12.1 12.6 12.8 13.7 14.1 14.2 17.1 18.3 19.7   
## 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.

unique(liver$Direct\_Bilirubin)

## [1] 0.1 5.5 4.1 0.4 2.0 0.7 0.2 0.3 1.3 0.8 0.5 1.0 3.0 1.9 1.2  
## [16] 7.8 0.6 1.1 3.2 1.8 8.8 1.6 4.5 2.8 4.0 2.7 2.4 1.5 2.3 3.6  
## [31] 6.2 7.0 8.2 11.3 10.2 2.5 1.4 1.7 5.6 2.2 2.1 4.9 5.0 0.9 12.6  
## [46] 7.6 9.0 4.6 11.8 14.2 8.9 6.4 9.5 3.3 11.4 4.3 3.7 2.6 3.9 5.1  
## [61] 12.8 10.4 17.1 14.1 8.5 10.0 12.1 2.9 5.2 18.3 7.2 11.7 10.8 6.1 4.2  
## [76] 19.7 7.7 8.4 6.0 13.7

length(unique(liver$Direct\_Bilirubin))

## [1] 80

min(liver$Direct\_Bilirubin)

## [1] 0.1

max(liver$Direct\_Bilirubin)

## [1] 19.7

Here the total observation of Age is 579 and is a continuous variable that range lies between 0.1-19.7. Now, we will see the summary of the Age variable and also age is a continues variable, for further analysis as follows:

summary(liver$Direct\_Bilirubin)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.100 0.200 0.300 1.494 1.300 19.700

sd(liver$Direct\_Bilirubin)

## [1] 2.816499

var(liver$Direct\_Bilirubin)

## [1] 7.932664

getMode <- function(v) {  
 uniqv <- unique(v)  
 uniqv[which.max(tabulate(match(v, uniqv)))]  
}  
v <- c(liver$Direct\_Bilirubin)  
mode <- getMode(v)  
print(mode)

## [1] 0.2

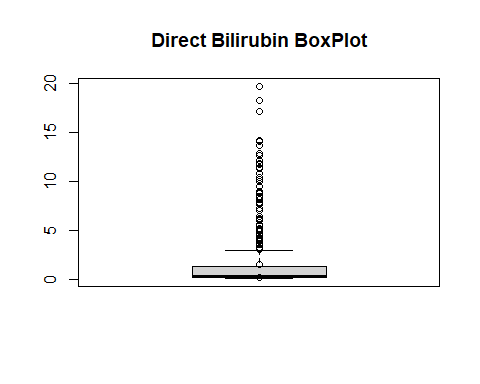
From the above summary we can observe that the Mean (1.494), Median (0.300) and Mode (0.2) are not equal so the distribution is asymmetrical. Here Mean > Median > Mode so the distribution could be positive and skewed to the right. Now we also confirm this as follows:

library(moments)  
skewness(liver$Direct\_Bilirubin)

## [1] 3.190869

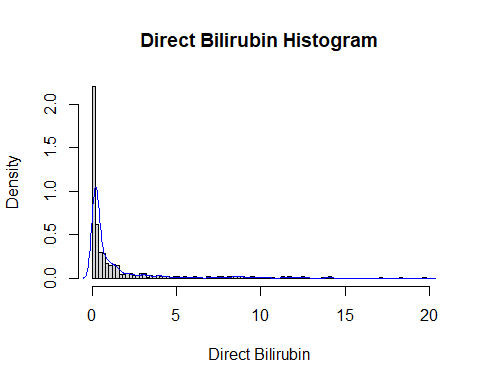
Hence, the skewness is the positive so the distribution is positively skewed was rightly observed. Also I will plot the Box plot to comparing the distribution of data across data set as follows:

boxplot(liver$Direct\_Bilirubin, main = "Direct Bilirubin BoxPlot")  
points(mean(liver$Direct\_Bilirubin))  
points(mode)



Box plots are graphical displays of the five number summary, So they describe the Minimum (lowest mean relative) lower whisker, Quartile 1 (It separates the lowest 25% observation to the rest of the observations), Median (Which divides the lower 50% observation from the upper 50% observations), Quartile 3 (It divides the upper 25% observation to the rest of the observations) and Maximum. In the above Boxplot diagram, there are many outliers and the upper whisker shows the highest mean relative. Now, I will plot a histogram a graphical display of data using bars of different heights to measures distribution of continuous data as follows:

hist(liver$Direct\_Bilirubin, prob = TRUE, breaks = 80, xlab = "Direct Bilirubin", main = "Direct Bilirubin Histogram")  
lines(density(liver$Direct\_Bilirubin), col='blue')



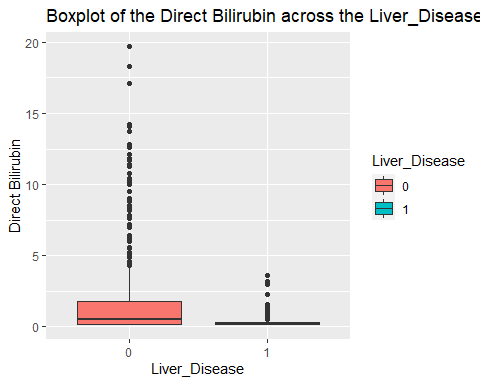
From the above histogram, we can observe that the diagram has many peaks.Using the density lines, I have approximated the histogram graph. Density provides information about the type of distribution under consideration and allows us to determine whether the attribute is distributed normally or not. From the graph above we can see how the Direct Bilirubin variable does not seem to fit to a Normal distribution, there is a peak of the frequency distributions around many values and also a left skewed. We have already observed a positive value of 3.190869, so we will affirm that the distribution is right skewed. To check whether the distribution is Platykurtic or not, we will check this by following R method as follows:

kurtosis(liver$Direct\_Bilirubin)

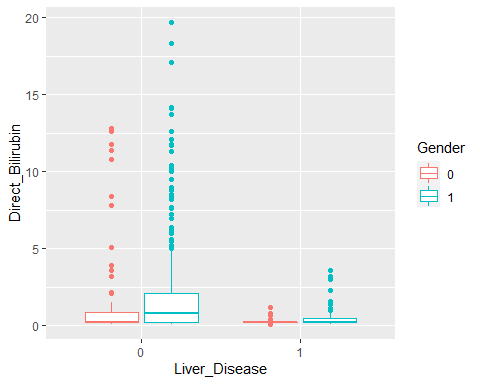
## [1] 14.14493

The distribution is also Leptokurtic, since the value is greater than 3.

ggplot(liver, aes(x = Liver\_Disease, y = Direct\_Bilirubin, fill = Liver\_Disease)) +  
 geom\_boxplot() +  
 ylab("Direct Bilirubin") +  
 ggtitle("Boxplot of the Direct Bilirubin across the Liver\_Disease")



ggplot(liver, aes(Liver\_Disease, Direct\_Bilirubin)) +   
 geom\_boxplot(aes(color = Gender))

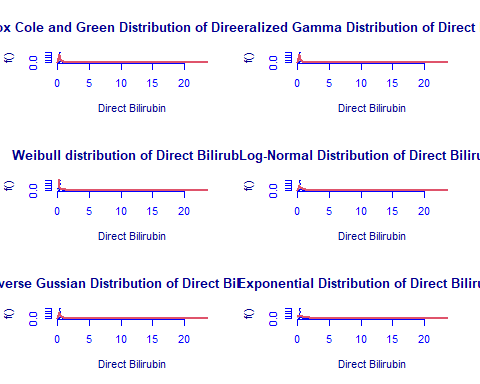


Similar to the Total Bilirubin we have skewed data for the Direct Bilirubin. The mean is 1.486, with a maximum being 19.7 and the median being 0.3. There is a difference in the mean and range of the Direct Bilirubin for the Liver\_Disease. There are differences also when you look t it further by gender. The mean Direct Bilirubin is higher for the male group across the Liver\_Disease. There is a bigger range for both genders for Direct Bilirubin when the response is 0.

### Direct Bilirubin Fit for the Data

Now we will try to fit different models to Age distribution evaluating the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC), The lower are the indexes, the better is the fitting. I will evaluate both the single parameter distributions and mixture distributions as follows:

library(MASS)  
library(gamlss)  
  
  
par(mfrow=c(3,2))  
  
db.BCCG <- histDist(liver$Direct\_Bilirubin, family=BCCG, nbins = 80, xlab = "Direct Bilirubin", main="Box-Cox Cole and Green Distribution of Direct Bilirubin")  
  
db.GG <- histDist(liver$Direct\_Bilirubin, family=GG, nbins = 80, xlab = "Direct Bilirubin", main="Generalized Gamma Distribution of Direct Bilirubin")  
  
db.WEI <- histDist(liver$Direct\_Bilirubin, family=WEI, nbins = 80, xlab = "Direct Bilirubin", main="Weibull distribution of Direct Bilirubin")  
  
db.LOGNO <- histDist(liver$Direct\_Bilirubin, family=LOGNO, nbins = 80, xlab = "Direct Bilirubin", main="Log-Normal Distribution of Direct Bilirubin")  
  
db.IG <- histDist(liver$Direct\_Bilirubin, family=IG, nbins = 80, xlab = "Direct Bilirubin", main = "Inverse Gussian Distribution of Direct Bilirubin")  
  
db.EXP<- histDist(liver$Direct\_Bilirubin, family=EXP, nbins = 80, xlab = "Direct Bilirubin", main = "Exponential Distribution of Direct Bilirubin")



library(Matrix)  
library(glmnet)  
df <- data.frame(Rownames = c("Box-Cox Cole and Green","Generalized Gamma", "Weibull",  
 "Log-Normal", "Inverse Gussian", "Exponential"),  
 AIC = c(AIC(db.BCCG), AIC(db.GG), AIC(db.WEI), AIC(db.LOGNO),  
 AIC(db.IG), AIC(db.EXP)),  
 BIC = c(db.BCCG$sbc, db.GG$sbc, db.WEI$sbc, db.LOGNO$sbc,  
 db.IG$sbc, db.EXP$sbc),  
 df = c(db.BCCG$df.fit, db.GG$df.fit, db.WEI$df.fit,  
 db.LOGNO$df.fit, db.IG$df.fit, db.EXP$df.fit),  
 LogLike = c(logLik(db.BCCG), logLik(db.GG), logLik(db.WEI),  
 logLik(db.LOGNO), logLik(db.IG), logLik(db.EXP)))  
df

## Rownames AIC BIC df LogLike  
## 1 Box-Cox Cole and Green 1078.120 1091.204 3 -536.0602  
## 2 Generalized Gamma 1062.880 1075.964 3 -528.4398  
## 3 Weibull 1417.420 1426.142 2 -706.7099  
## 4 Log-Normal 1227.517 1236.240 2 -611.7586  
## 5 Inverse Gussian 1119.812 1128.535 2 -557.9061  
## 6 Exponential 1624.986 1629.348 1 -811.4932

As we can see, the model with the highest log likelihood (-528.4398) and the lowest AIC (1062.880) and BIC (1075.964) is the Generalized Gamma Distribution. Therefore, based on the greatest likelihood technique, our data fits better in the Generalized Gamma Distribution. Now, I will also compare two models by means of the Likelihood ratio test as we performed above as follows:

library(zoo)  
library(lmtest)  
lrtest(db.GG, db.EXP)

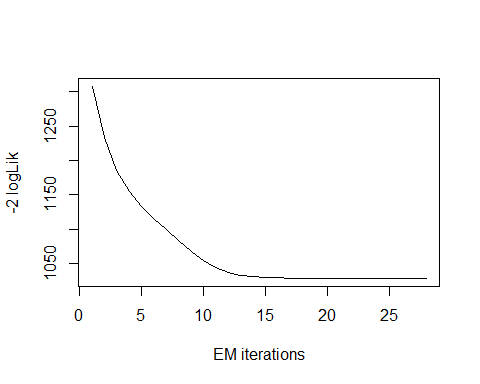
## Likelihood ratio test  
##   
## Model 1: gamlssML(formula = liver$Direct\_Bilirubin, family = "GG")  
## Model 2: gamlssML(formula = liver$Direct\_Bilirubin, family = "EXP")  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -528.44   
## 2 1 -811.49 -2 566.11 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Under the null hypothesis, we compare the Generalized Gamma distribution to the Exponential distribution under the alternative hypothesis. At any level of significance, we can reject the null hypothesis. Because our distributions fit better in the Generalized Gamma model, we’ll use it.

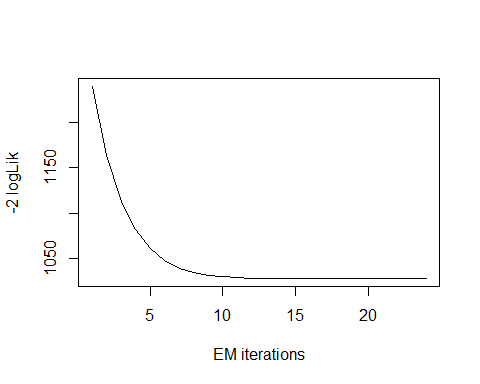
### Distributions Mixture

Now, I will also apply the fitting criteria for the distributions with Gamma mixture as follows:

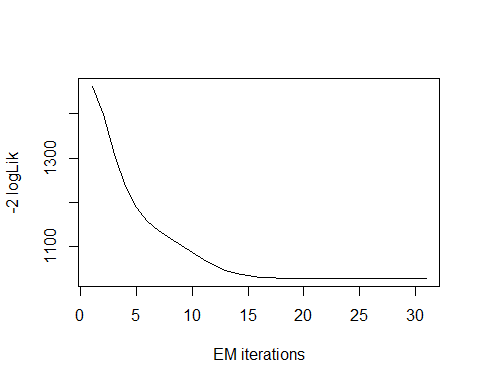
library(gamlss.mx)  
mix.gam <- gamlssMXfits(n = 5, liver$Direct\_Bilirubin~1, family = GA, K = 2, data = liver)



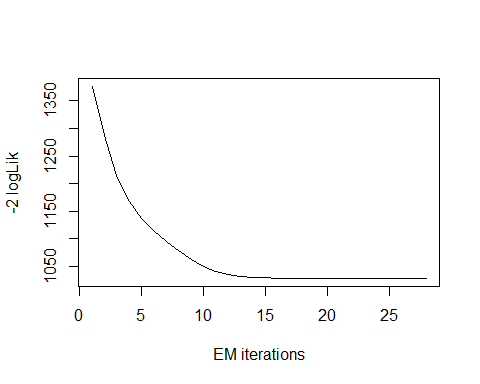
## model= 1



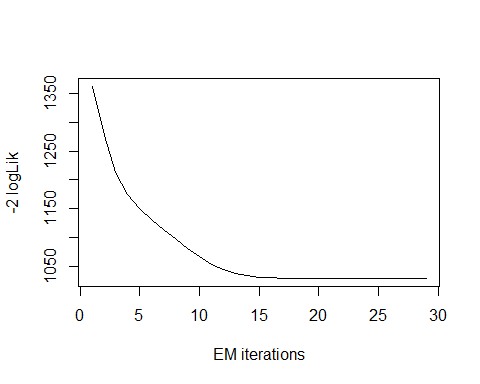
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam)

##   
## Mixing Family: c("GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Direct\_Bilirubin ~ 1, family = GA,   
## K = 2, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## -1.587   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -1.058   
## Mu Coefficients for model: 2   
## (Intercept)   
## 0.9989   
## Sigma Coefficients for model: 2   
## (Intercept)   
## 0.08336   
##   
## Estimated probabilities: 0.4863282 0.5136718   
##   
## Degrees of Freedom for the fit: 5 Residual Deg. of Freedom 574   
## Global Deviance: 1028.49   
## AIC: 1038.49   
## SBC: 1060.3

We can observe that the AIC value of the mixture of Gamma has improved, since it is lower than that of the single Gamma distribution. The current AIC value is 1038.49, whereas the previous value was 1062.880 and the current BIC value is 1060.3, whereas the previous value was 1075.964.

logLik(mix.gam)

## 'log Lik.' -514.2449 (df=5)

mix.gam$prob

## [1] 0.4863282 0.5136718

fitted(mix.gam, "mu")[1]

## [1] 1.494229

fitted(mix.gam, "sigma")[2]

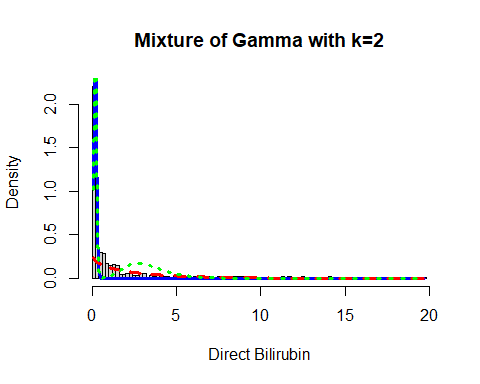
## [1] 1.494229

hist(liver$Direct\_Bilirubin, breaks = 80, xlab = "Direct Bilirubin", main="Mixture of Gamma with k=2", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Direct\_Bilirubin, breaks = 80, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Direct\_Bilirubin, breaks = 80, freq = FALSE, :  
## argument 'freq' is not made use of

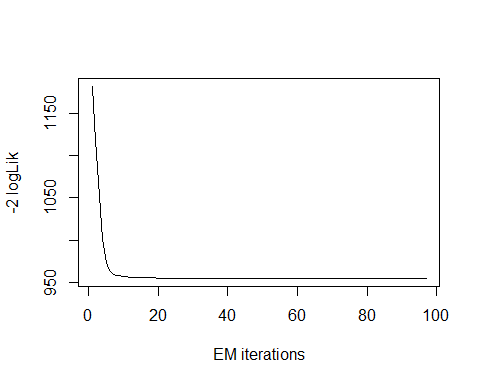
## $breaks  
## [1] 0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8  
## [16] 3.0 3.2 3.4 3.6 3.8 4.0 4.2 4.4 4.6 4.8 5.0 5.2 5.4 5.6 5.8  
## [31] 6.0 6.2 6.4 6.6 6.8 7.0 7.2 7.4 7.6 7.8 8.0 8.2 8.4 8.6 8.8  
## [46] 9.0 9.2 9.4 9.6 9.8 10.0 10.2 10.4 10.6 10.8 11.0 11.2 11.4 11.6 11.8  
## [61] 12.0 12.2 12.4 12.6 12.8 13.0 13.2 13.4 13.6 13.8 14.0 14.2 14.4 14.6 14.8  
## [76] 15.0 15.2 15.4 15.6 15.8 16.0 16.2 16.4 16.6 16.8 17.0 17.2 17.4 17.6 17.8  
## [91] 18.0 18.2 18.4 18.6 18.8 19.0 19.2 19.4 19.6 19.8  
##   
## $counts  
## [1] 255 71 35 33 20 17 19 17 5 5 6 6 4 4 6 6 1 4 2  
## [20] 4 3 2 3 0 3 2 0 2 0 1 2 1 0 0 2 1 0 2  
## [39] 2 0 2 2 2 2 3 0 0 3 0 1 1 1 0 1 0 0 2  
## [58] 0 3 0 1 0 2 1 0 0 0 0 1 0 2 0 0 0 0 0  
## [77] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0  
## [96] 0 0 0 1  
##   
## $density  
## [1] 2.202072539 0.613126079 0.302245250 0.284974093 0.172711572 0.146804836  
## [7] 0.164075993 0.146804836 0.043177893 0.043177893 0.051813472 0.051813472  
## [13] 0.034542314 0.034542314 0.051813472 0.051813472 0.008635579 0.034542314  
## [19] 0.017271157 0.034542314 0.025906736 0.017271157 0.025906736 0.000000000  
## [25] 0.025906736 0.017271157 0.000000000 0.017271157 0.000000000 0.008635579  
## [31] 0.017271157 0.008635579 0.000000000 0.000000000 0.017271157 0.008635579  
## [37] 0.000000000 0.017271157 0.017271157 0.000000000 0.017271157 0.017271157  
## [43] 0.017271157 0.017271157 0.025906736 0.000000000 0.000000000 0.025906736  
## [49] 0.000000000 0.008635579 0.008635579 0.008635579 0.000000000 0.008635579  
## [55] 0.000000000 0.000000000 0.017271157 0.000000000 0.025906736 0.000000000  
## [61] 0.008635579 0.000000000 0.017271157 0.008635579 0.000000000 0.000000000  
## [67] 0.000000000 0.000000000 0.008635579 0.000000000 0.017271157 0.000000000  
## [73] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [79] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [85] 0.000000000 0.008635579 0.000000000 0.000000000 0.000000000 0.000000000  
## [91] 0.000000000 0.008635579 0.000000000 0.000000000 0.000000000 0.000000000  
## [97] 0.000000000 0.000000000 0.008635579  
##   
## $mids  
## [1] 0.1 0.3 0.5 0.7 0.9 1.1 1.3 1.5 1.7 1.9 2.1 2.3 2.5 2.7 2.9  
## [16] 3.1 3.3 3.5 3.7 3.9 4.1 4.3 4.5 4.7 4.9 5.1 5.3 5.5 5.7 5.9  
## [31] 6.1 6.3 6.5 6.7 6.9 7.1 7.3 7.5 7.7 7.9 8.1 8.3 8.5 8.7 8.9  
## [46] 9.1 9.3 9.5 9.7 9.9 10.1 10.3 10.5 10.7 10.9 11.1 11.3 11.5 11.7 11.9  
## [61] 12.1 12.3 12.5 12.7 12.9 13.1 13.3 13.5 13.7 13.9 14.1 14.3 14.5 14.7 14.9  
## [76] 15.1 15.3 15.5 15.7 15.9 16.1 16.3 16.5 16.7 16.9 17.1 17.3 17.5 17.7 17.9  
## [91] 18.1 18.3 18.5 18.7 18.9 19.1 19.3 19.5 19.7  
##   
## $xname  
## [1] "liver$Direct\_Bilirubin"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin), length = length(liver$Direct\_Bilirubin)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin),  
 length = length(liver$Direct\_Bilirubin)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin), length = length(liver$Direct\_Bilirubin)),  
 mix.gam[["prob"]][2]\*dGA(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin),  
 length = length(liver$Direct\_Bilirubin)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin), length = length(liver$Direct\_Bilirubin)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin),  
 length = length(liver$Direct\_Bilirubin)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam[["prob"]][2]\*dRG(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin),  
 length = length(liver$Direct\_Bilirubin)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")

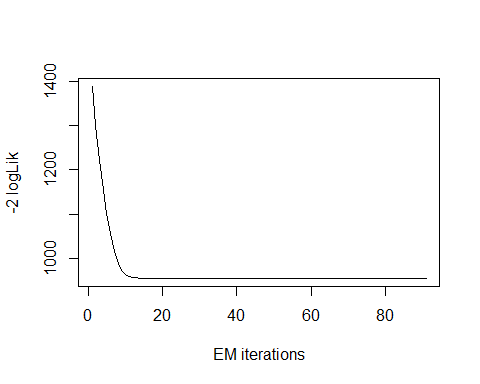


Now, I will again apply the fitting criteria for the distributions with Gamma mixture with the k = 3 as follows:

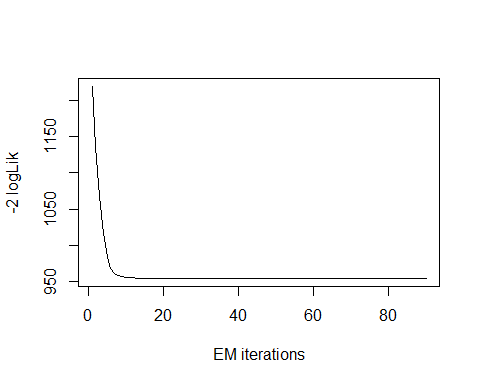
mix.gam.3 <- gamlssMXfits(n = 5, liver$Direct\_Bilirubin~1, family = GA, K = 3, data = liver)



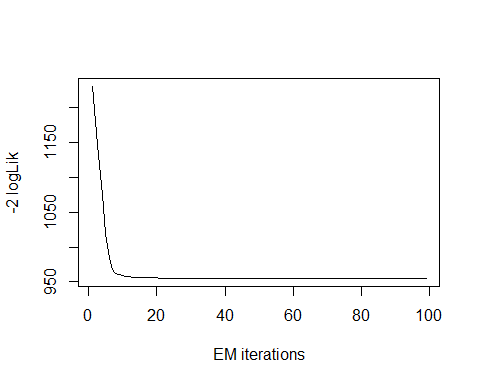
## model= 1



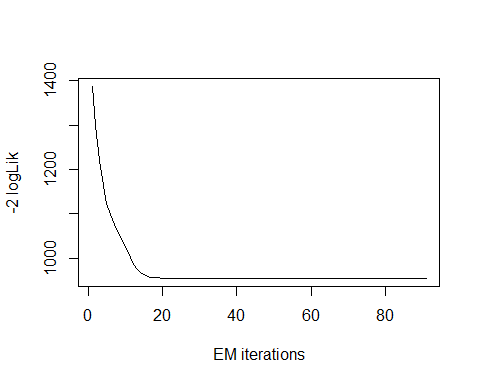
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam.3)

##   
## Mixing Family: c("GA", "GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Direct\_Bilirubin ~ 1, family = GA,   
## K = 3, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## -0.06072   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -0.4776   
## Mu Coefficients for model: 2   
## (Intercept)   
## 1.748   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -0.275   
## Mu Coefficients for model: 3   
## (Intercept)   
## -1.619   
## Sigma Coefficients for model: 3   
## (Intercept)   
## -1.065   
##   
## Estimated probabilities: 0.3069872 0.1924769 0.5005359   
##   
## Degrees of Freedom for the fit: 8 Residual Deg. of Freedom 571   
## Global Deviance: 954.941   
## AIC: 970.941   
## SBC: 1005.83

logLik(mix.gam.3)

## 'log Lik.' -477.4706 (df=8)

mix.gam.3$prob

## [1] 0.3069872 0.1924769 0.5005359

fitted(mix.gam.3, "mu")[1]

## [1] 1.493881

fitted(mix.gam.3, "sigma")[2]

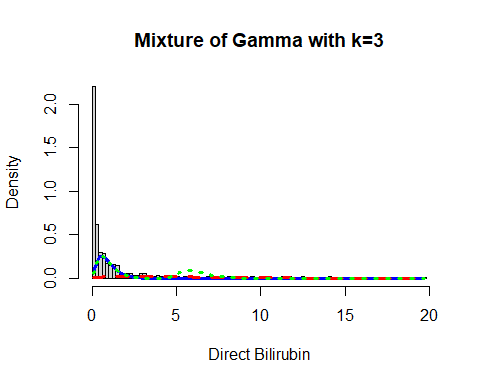
## [1] 1.493881

hist(liver$Direct\_Bilirubin, breaks = 80, xlab = "Direct Bilirubin", main="Mixture of Gamma with k=3", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam.3[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam.3[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam.3[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam.3[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Direct\_Bilirubin, breaks = 80, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Direct\_Bilirubin, breaks = 80, freq = FALSE, :  
## argument 'freq' is not made use of

## $breaks  
## [1] 0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8  
## [16] 3.0 3.2 3.4 3.6 3.8 4.0 4.2 4.4 4.6 4.8 5.0 5.2 5.4 5.6 5.8  
## [31] 6.0 6.2 6.4 6.6 6.8 7.0 7.2 7.4 7.6 7.8 8.0 8.2 8.4 8.6 8.8  
## [46] 9.0 9.2 9.4 9.6 9.8 10.0 10.2 10.4 10.6 10.8 11.0 11.2 11.4 11.6 11.8  
## [61] 12.0 12.2 12.4 12.6 12.8 13.0 13.2 13.4 13.6 13.8 14.0 14.2 14.4 14.6 14.8  
## [76] 15.0 15.2 15.4 15.6 15.8 16.0 16.2 16.4 16.6 16.8 17.0 17.2 17.4 17.6 17.8  
## [91] 18.0 18.2 18.4 18.6 18.8 19.0 19.2 19.4 19.6 19.8  
##   
## $counts  
## [1] 255 71 35 33 20 17 19 17 5 5 6 6 4 4 6 6 1 4 2  
## [20] 4 3 2 3 0 3 2 0 2 0 1 2 1 0 0 2 1 0 2  
## [39] 2 0 2 2 2 2 3 0 0 3 0 1 1 1 0 1 0 0 2  
## [58] 0 3 0 1 0 2 1 0 0 0 0 1 0 2 0 0 0 0 0  
## [77] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0  
## [96] 0 0 0 1  
##   
## $density  
## [1] 2.202072539 0.613126079 0.302245250 0.284974093 0.172711572 0.146804836  
## [7] 0.164075993 0.146804836 0.043177893 0.043177893 0.051813472 0.051813472  
## [13] 0.034542314 0.034542314 0.051813472 0.051813472 0.008635579 0.034542314  
## [19] 0.017271157 0.034542314 0.025906736 0.017271157 0.025906736 0.000000000  
## [25] 0.025906736 0.017271157 0.000000000 0.017271157 0.000000000 0.008635579  
## [31] 0.017271157 0.008635579 0.000000000 0.000000000 0.017271157 0.008635579  
## [37] 0.000000000 0.017271157 0.017271157 0.000000000 0.017271157 0.017271157  
## [43] 0.017271157 0.017271157 0.025906736 0.000000000 0.000000000 0.025906736  
## [49] 0.000000000 0.008635579 0.008635579 0.008635579 0.000000000 0.008635579  
## [55] 0.000000000 0.000000000 0.017271157 0.000000000 0.025906736 0.000000000  
## [61] 0.008635579 0.000000000 0.017271157 0.008635579 0.000000000 0.000000000  
## [67] 0.000000000 0.000000000 0.008635579 0.000000000 0.017271157 0.000000000  
## [73] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [79] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000  
## [85] 0.000000000 0.008635579 0.000000000 0.000000000 0.000000000 0.000000000  
## [91] 0.000000000 0.008635579 0.000000000 0.000000000 0.000000000 0.000000000  
## [97] 0.000000000 0.000000000 0.008635579  
##   
## $mids  
## [1] 0.1 0.3 0.5 0.7 0.9 1.1 1.3 1.5 1.7 1.9 2.1 2.3 2.5 2.7 2.9  
## [16] 3.1 3.3 3.5 3.7 3.9 4.1 4.3 4.5 4.7 4.9 5.1 5.3 5.5 5.7 5.9  
## [31] 6.1 6.3 6.5 6.7 6.9 7.1 7.3 7.5 7.7 7.9 8.1 8.3 8.5 8.7 8.9  
## [46] 9.1 9.3 9.5 9.7 9.9 10.1 10.3 10.5 10.7 10.9 11.1 11.3 11.5 11.7 11.9  
## [61] 12.1 12.3 12.5 12.7 12.9 13.1 13.3 13.5 13.7 13.9 14.1 14.3 14.5 14.7 14.9  
## [76] 15.1 15.3 15.5 15.7 15.9 16.1 16.3 16.5 16.7 16.9 17.1 17.3 17.5 17.7 17.9  
## [91] 18.1 18.3 18.5 18.7 18.9 19.1 19.3 19.5 19.7  
##   
## $xname  
## [1] "liver$Direct\_Bilirubin"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin), length = length(liver$Direct\_Bilirubin)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin),  
 length = length(liver$Direct\_Bilirubin)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin), length = length(liver$Direct\_Bilirubin)),  
 mix.gam.3[["prob"]][2]\*dGA(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin),  
 length = length(liver$Direct\_Bilirubin)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin), length = length(liver$Direct\_Bilirubin)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin),  
 length = length(liver$Direct\_Bilirubin)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam.3[["prob"]][2]\*dRG(seq(min(liver$Direct\_Bilirubin), max(liver$Direct\_Bilirubin),  
 length = length(liver$Direct\_Bilirubin)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")



mix.gm.tb <- data.frame(row.names=c('Gamma Mixture, K=3','Gamma Mixture, K=2', "Generalized Gamma"),  
 AIC=c(mix.gam.3$aic, mix.gam$aic, AIC(age.GG)),  
 BIC=c(mix.gam.3$sbc, mix.gam$sbc, age.GG$sbc))  
mix.gm.tb

## AIC BIC  
## Gamma Mixture, K=3 970.9412 1005.832  
## Gamma Mixture, K=2 1038.4897 1060.296  
## Generalized Gamma 4862.3309 4875.415

We can observe that the AIC value of the mixture of Gamma with k=3 and k=2 has improved, since values are lower than that of the single Gamma distribution. The previous AIC value of k=2 is 1038.4897, whereas the current value which is lower is 970.9413 and the previous BIC value of k=2 was 1060.296, whereas the current value which is lower is 1005.832. Here we can clearly see that our data fits better in the Gamma Mixture Distribution with k=3.

## Alkaline Phosphotase

Lets explore the Alkaline Phosphotase variable:

head(liver$Alkaline\_Phosphotase)

## [1] 187 699 490 182 195 208

length(liver$Alkaline\_Phosphotase)

## [1] 579

table(liver$Alkaline\_Phosphotase)

##   
## 63 75 90 92 97 98 100 102 103 105 108 110 114 115 116 120   
## 1 1 1 2 1 1 2 1 1 1 1 2 1 1 1 1   
## 123 125 127 128 130 134 135 137 138 140 142 143 144 145 146 147   
## 1 1 1 1 2 1 2 2 1 4 3 1 2 9 3 1   
## 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163   
## 2 1 2 1 5 1 2 2 2 3 9 5 5 2 6 3   
## 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179   
## 1 8 1 1 6 1 4 4 2 3 3 6 3 1 5 2   
## 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195   
## 9 1 9 2 1 5 5 2 7 5 10 3 6 1 5 10   
## 196 197 198 199 200 201 202 204 205 206 208 209 210 211 212 214   
## 7 1 11 2 2 4 7 2 6 6 5 2 4 1 1 4   
## 215 216 218 219 220 224 225 226 227 228 230 231 232 234 235 236   
## 11 3 8 2 3 3 1 1 1 1 5 1 1 1 1 2   
## 237 238 239 240 243 245 246 247 248 250 251 253 254 256 257 258   
## 2 3 3 2 1 3 1 1 1 2 1 1 1 1 2 3   
## 259 260 262 263 265 268 269 270 271 272 275 276 279 280 282 285   
## 1 2 1 2 3 2 1 1 1 5 3 1 2 2 8 4   
## 286 289 290 292 293 298 300 302 305 308 309 310 312 314 315 316   
## 1 4 6 2 2 11 3 1 1 1 1 6 1 1 3 2   
## 320 326 331 332 335 340 342 348 349 350 352 356 358 360 365 367   
## 2 1 1 3 1 1 1 1 1 3 1 1 3 1 1 1   
## 374 375 380 386 388 390 392 395 400 401 405 406 410 415 418 430   
## 1 1 3 1 1 1 1 1 1 1 1 1 2 1 1 1   
## 450 458 460 462 466 470 480 482 486 490 498 500 505 509 512 515   
## 2 1 1 1 1 1 1 3 1 2 1 1 1 1 1 1   
## 518 527 538 542 554 555 558 560 562 574 575 580 588 592 599 610   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 612 614 621 630 650 661 664 670 680 686 690 699 719 750 768 802   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1   
## 805 850 859 862 901 915 950 962 1020 1050 1100 1110 1124 1350 1420 1550   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1   
## 1580 1620 1630 1750 1896 2110   
## 1 1 1 1 1 1

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.

unique(liver$Alkaline\_Phosphotase)

## [1] 187 699 490 182 195 208 154 202 290 210 260 310 214 145 183  
## [16] 342 165 293 610 482 542 231 194 289 240 128 188 190 156 410  
## [31] 374 263 275 168 160 630 415 150 230 176 206 170 161 253 198  
## [46] 272 175 367 158 259 470 215 239 186 205 171 162 518 1620 146  
## [61] 670 915 75 148 258 237 269 320 298 538 238 308 204 282 265  
## [76] 312 243 224 225 486 257 179 661 1580 1630 280 300 178 177 201  
## [91] 802 248 1896 512 199 1110 380 159 332 189 392 286 180 218 462  
## [106] 196 750 1050 599 292 962 950 200 1020 562 386 250 191 614 314  
## [121] 209 1124 664 142 169 1420 135 163 285 350 220 219 401 100 116  
## [136] 125 147 192 400 120 173 157 2110 360 316 498 480 680 152 859  
## [151] 901 335 245 505 228 185 247 348 140 358 110 235 460 262 144  
## [166] 123 575 155 315 174 340 234 430 588 527 574 216 63 302 211  
## [181] 458 375 405 650 115 621 256 418 271 130 558 326 331 172 105  
## [196] 102 149 580 92 719 554 555 509 690 862 592 450 1350 246 166  
## [211] 1750 236 212 279 181 1550 1100 686 309 164 270 137 90 167 197  
## [226] 226 352 103 850 276 193 805 151 349 365 305 127 254 108 268  
## [241] 138 466 227 395 97 406 114 153 768 232 390 356 388 143 251  
## [256] 134 612 515 560 500 98 184

length(unique(liver$Alkaline\_Phosphotase))

## [1] 262

min(liver$Alkaline\_Phosphotase)

## [1] 63

max(liver$Alkaline\_Phosphotase)

## [1] 2110

Here the total observation of Age is 579 and is a continuous variable that range lies between 0.08-82. Now, we will see the summary of the Age variable and also age is a continues variable, for further analysis as follows:

summary(liver$Alkaline\_Phosphotase)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 63.0 175.5 208.0 291.4 298.0 2110.0

sd(liver$Alkaline\_Phosphotase)

## [1] 243.5619

var(liver$Alkaline\_Phosphotase)

## [1] 59322.38

v <- c(liver$Alkaline\_Phosphotase)  
mode <- getMode(v)  
print(mode)

## [1] 198

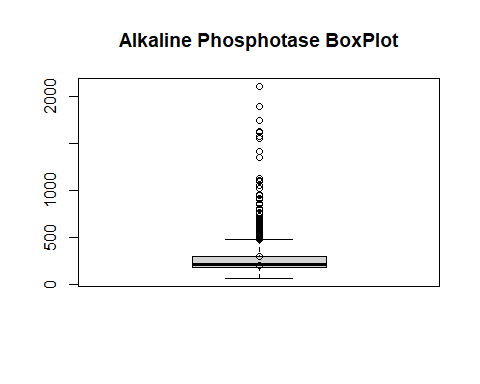
From the above summary we can observe that the Mean (291.4), Median (208.0) and Mode (198) are not equal so the distribution is asymmetrical. Here Mean > Median > Mode so the distribution could be positive and skewed to the right. Now we also confirm this as follows:

library(moments)  
skewness(liver$Alkaline\_Phosphotase)

## [1] 3.743771

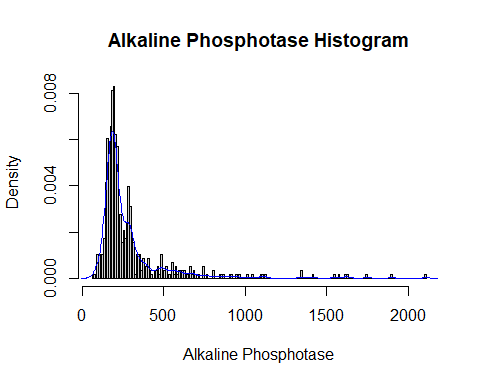
Hence, the skewness is the positive so the distribution is positively skewed was rightly observed. Also I will plot the Boxplot to comparing the distribution of data across dataset as follows:

boxplot(liver$Alkaline\_Phosphotase, main = "Alkaline Phosphotase BoxPlot")  
points(mean(liver$Alkaline\_Phosphotase))  
points(mode)



Box plots are graphical displays of the five number summary, So they describe the Minimum (lowest mean relative) lower whisker, Quartile 1 (It separates the lowest 25% observation to the rest of the observations), Median (Which divides the lower 50% observation from the upper 50% observations), Quartile 3 (It divides the upper 25% observation to the rest of the observations) and Maximum. In the above Box plot diagram, there are many outliers and the upper whisker shows the highest mean relative. Now, I will plot a histogram a graphical display of data using bars of different heights to measures distribution of continuous data as follows:

hist(liver$Alkaline\_Phosphotase, prob = TRUE, breaks = 262, xlab = "Alkaline Phosphotase", main = "Alkaline Phosphotase Histogram")  
lines(density(liver$Alkaline\_Phosphotase), col='blue')



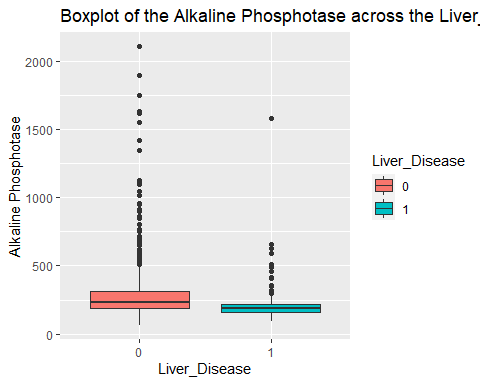
From the above histogram, we can observe that the diagram has many peaks.Using the density lines, I have approximated the histogram graph. Density provides information about the type of distribution under consideration and allows us to determine whether the attribute is distributed normally or not. From the graph above we can see how the Alkaline\_Phosphotase variable does not seem to fit to a Normal distribution, there is a peak of the frequency distributions around many values and also a right skewed. We have already observed a positive value of 3.743771, so we will affirm that the distribution is right skewed. To check whether the distribution is Platykurtic or not, we will check this by following R method as follows:

kurtosis(liver$Alkaline\_Phosphotase)

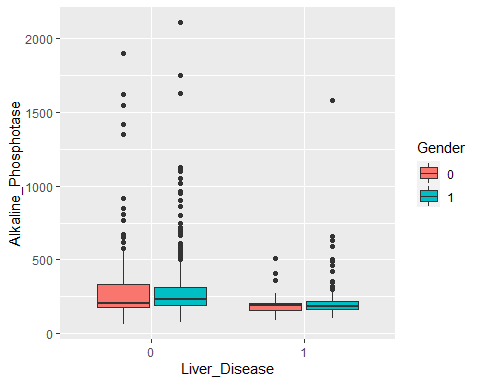
## [1] 20.47242

The distribution is also Leptokurtic, since the value is greater than 3.

ggplot(liver, aes(x = Liver\_Disease, y = Alkaline\_Phosphotase, fill = Liver\_Disease)) +  
 geom\_boxplot() +  
 ylab("Alkaline Phosphotase") +  
 ggtitle("Boxplot of the Alkaline Phosphotase across the Liver\_Disease")



ggplot(liver, aes(Liver\_Disease, Alkaline\_Phosphotase)) +   
 geom\_boxplot(aes(fill = Gender))

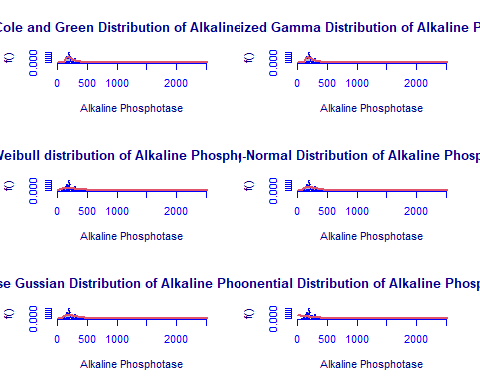


The range for the Alkaline Phosphotase is wide, with a mean of 290.6, median or 208, minimum of 63 and maximum of 2110. The numbers for Alkaline Phosphotase are also skewed as shown in the plot. When you look at the numbers across the response we see a difference across it. The mean is higher when the Liver\_Disease is 0 and the range is also wider for Liver\_Disease = 0. When you look further and compare by gender, there are differences in the mean within the Liver\_Disease = 0, but not much for the Liver\_Disease = 1 group.

### Alkaline Phosphotase Fit for the Data

Now we will try to fit different models to Alkaline Phosphotase distribution evaluating the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC), The lower are the indexes, the better is the fitting. I will evaluate both the single parameter distributions and mixture distributions as follows:

library(MASS)  
library(gamlss)  
  
  
par(mfrow=c(3,2))  
  
ap.BCCG <- histDist(liver$Alkaline\_Phosphotase, family=BCCG, nbins = 262, xlab = "Alkaline Phosphotase", main="Box-Cox Cole and Green Distribution of Alkaline Phosphotase")  
  
ap.GG <- histDist(liver$Alkaline\_Phosphotase, family=GG, nbins = 262, xlab = "Alkaline Phosphotase", main="Generalized Gamma Distribution of Alkaline Phosphotase")  
  
ap.WEI <- histDist(liver$Alkaline\_Phosphotase, family=WEI, nbins = 262, xlab = "Alkaline Phosphotase", main="Weibull distribution of Alkaline Phosphotase")  
  
ap.LOGNO <- histDist(liver$Alkaline\_Phosphotase, family=LOGNO, nbins = 262, xlab = "Alkaline Phosphotase", main="Log-Normal Distribution of Alkaline Phosphotase")  
  
ap.IG <- histDist(liver$Alkaline\_Phosphotase, family=IG, nbins=262, xlab = "Alkaline Phosphotase", main = "Inverse Gussian Distribution of Alkaline Phosphotase")  
  
ap.EXP<- histDist(liver$Alkaline\_Phosphotase, family=EXP, nbins=262, xlab = "Alkaline Phosphotase", main = "Exponential Distribution of Alkaline Phosphotase")



library(Matrix)  
library(glmnet)  
df <- data.frame(Rownames = c("Box-Cox Cole and Green","Generalized Gamma", "Weibull",  
 "Log-Normal", "Inverse Gussian", "Exponential"),  
 AIC = c(AIC(ap.BCCG), AIC(ap.GG), AIC(ap.WEI), AIC(ap.LOGNO),  
 AIC(ap.IG), AIC(ap.EXP)),  
 BIC = c(ap.BCCG$sbc, ap.GG$sbc, ap.WEI$sbc, ap.LOGNO$sbc,  
 ap.IG$sbc, ap.EXP$sbc),  
 df = c(ap.BCCG$df.fit, ap.GG$df.fit, ap.WEI$df.fit,  
 ap.LOGNO$df.fit, ap.IG$df.fit, ap.EXP$df.fit),  
 LogLike = c(logLik(ap.BCCG), logLik(ap.GG), logLik(ap.WEI),  
 logLik(ap.LOGNO), logLik(ap.IG), logLik(ap.EXP)))  
df

## Rownames AIC BIC df LogLike  
## 1 Box-Cox Cole and Green 7132.341 7145.425 3 -3563.171  
## 2 Generalized Gamma 7140.210 7153.294 3 -3567.105  
## 3 Weibull 7580.311 7589.034 2 -3788.156  
## 4 Log-Normal 7272.052 7280.774 2 -3634.026  
## 5 Inverse Gussian 7282.613 7291.335 2 -3639.306  
## 6 Exponential 7731.164 7735.526 1 -3864.582

As we can see, the model with the highest log likelihood (-3563.171) and the lowest AIC (7132.341) and BIC (7145.425) is the Box-Cox Cole and Green Distribution. Therefore, based on the greatest likelihood technique, our data fits better in the Box-Cox Cole and Green Distribution. Now, I will also compare two models by means of the Likelihood ratio test as we performed above as follows:

library(zoo)  
library(lmtest)  
lrtest(ap.BCCG, ap.EXP)

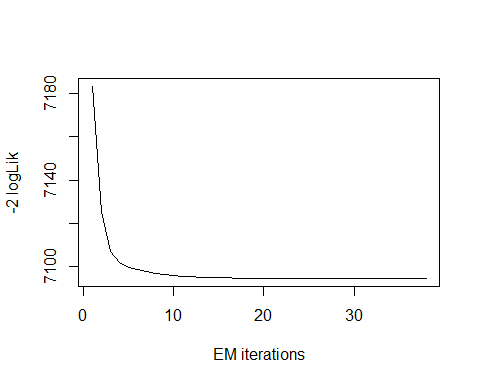
## Likelihood ratio test  
##   
## Model 1: gamlssML(formula = liver$Alkaline\_Phosphotase, family = "BCCG")  
## Model 2: gamlssML(formula = liver$Alkaline\_Phosphotase, family = "EXP")  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -3563.2   
## 2 1 -3864.6 -2 602.82 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Under the null hypothesis, we compare the Box-Cox Cole and Green Distribution to the Exponential distribution under the alternative hypothesis. At any level of significance, we can reject the null hypothesis. Because our distributions fit better in the Box-Cox Cole and Green model, we’ll use it.

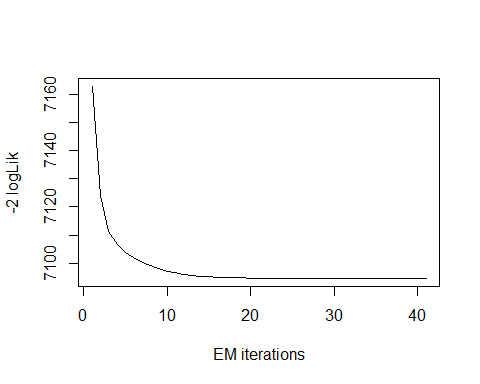
### Distributions Mixture

Now, I will also apply the fitting criteria for the distributions with Gamma mixture as follows:

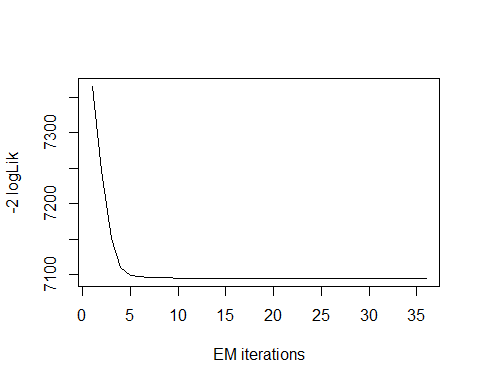
library(gamlss.mx)  
mix.gam <- gamlssMXfits(n = 5, liver$Alkaline\_Phosphotase~1, family = GA, K = 2, data = liver)



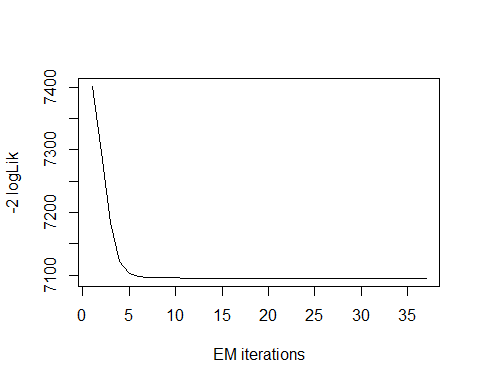
## model= 1



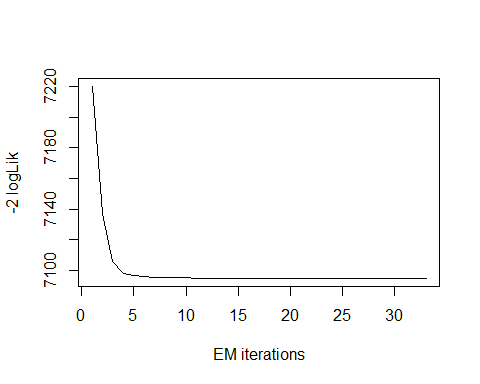
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam)

##   
## Mixing Family: c("GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Alkaline\_Phosphotase ~ 1,   
## family = GA, K = 2, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 5.329   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -1.339   
## Mu Coefficients for model: 2   
## (Intercept)   
## 6.285   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -0.4683   
##   
## Estimated probabilities: 0.7422594 0.2577406   
##   
## Degrees of Freedom for the fit: 5 Residual Deg. of Freedom 574   
## Global Deviance: 7094.69   
## AIC: 7104.69   
## SBC: 7126.5

We can observe that the AIC value of the mixture of Gamma has improved, since it is lower than that of the Box-Cox Cole and Green Distribution. The current AIC value is 7104.69, whereas the previous value was 7132.341 and the current value of BIC is 7126.5 whereas the previous value was 7145.425.

logLik(mix.gam)

## 'log Lik.' -3547.345 (df=5)

mix.gam$prob

## [1] 0.7422594 0.2577406

fitted(mix.gam, "mu")[1]

## [1] 291.3451

fitted(mix.gam, "sigma")[2]

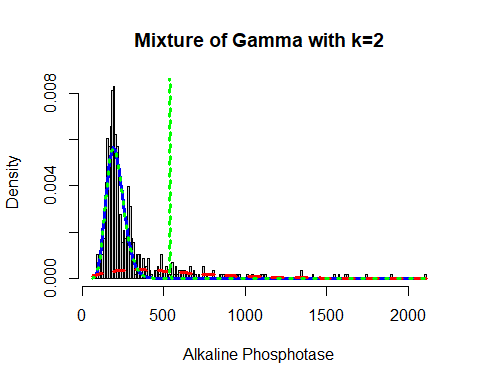
## [1] 291.3451

hist(liver$Alkaline\_Phosphotase, breaks = 262, xlab = "Alkaline Phosphotase", main="Mixture of Gamma with k=2", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Alkaline\_Phosphotase, breaks = 262, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Alkaline\_Phosphotase, breaks = 262, freq =  
## FALSE, : argument 'freq' is not made use of

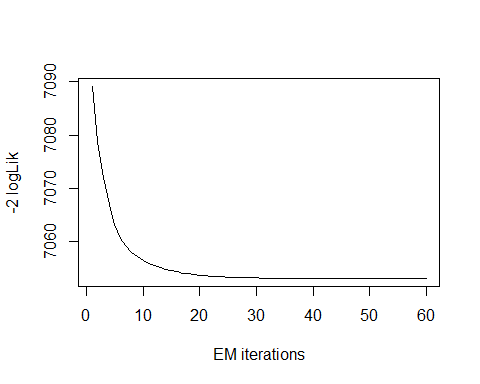
## $breaks  
## [1] 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200  
## [16] 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350  
## [31] 360 370 380 390 400 410 420 430 440 450 460 470 480 490 500  
## [46] 510 520 530 540 550 560 570 580 590 600 610 620 630 640 650  
## [61] 660 670 680 690 700 710 720 730 740 750 760 770 780 790 800  
## [76] 810 820 830 840 850 860 870 880 890 900 910 920 930 940 950  
## [91] 960 970 980 990 1000 1010 1020 1030 1040 1050 1060 1070 1080 1090 1100  
## [106] 1110 1120 1130 1140 1150 1160 1170 1180 1190 1200 1210 1220 1230 1240 1250  
## [121] 1260 1270 1280 1290 1300 1310 1320 1330 1340 1350 1360 1370 1380 1390 1400  
## [136] 1410 1420 1430 1440 1450 1460 1470 1480 1490 1500 1510 1520 1530 1540 1550  
## [151] 1560 1570 1580 1590 1600 1610 1620 1630 1640 1650 1660 1670 1680 1690 1700  
## [166] 1710 1720 1730 1740 1750 1760 1770 1780 1790 1800 1810 1820 1830 1840 1850  
## [181] 1860 1870 1880 1890 1900 1910 1920 1930 1940 1950 1960 1970 1980 1990 2000  
## [196] 2010 2020 2030 2040 2050 2060 2070 2080 2090 2100 2110  
##   
## $counts  
## [1] 1 1 1 6 6 4 6 10 24 35 33 38 47 48 36 33 12 16 9 12 10 14 23 18 10  
## [26] 9 1 6 6 6 2 5 3 3 5 2 1 0 2 2 3 1 6 2 2 3 1 1 1 4  
## [51] 1 3 1 2 1 2 2 0 1 0 3 1 2 1 0 1 0 0 3 0 1 0 0 0 2  
## [76] 0 0 0 1 1 1 0 0 0 1 1 0 0 1 0 1 0 0 0 0 1 0 0 1 0  
## [101] 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [126] 0 0 0 2 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0  
## [151] 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0  
## [176] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [201] 0 0 0 0 1  
##   
## $density  
## [1] 0.0001727116 0.0001727116 0.0001727116 0.0010362694 0.0010362694  
## [6] 0.0006908463 0.0010362694 0.0017271157 0.0041450777 0.0060449050  
## [11] 0.0056994819 0.0065630397 0.0081174439 0.0082901554 0.0062176166  
## [16] 0.0056994819 0.0020725389 0.0027633851 0.0015544041 0.0020725389  
## [21] 0.0017271157 0.0024179620 0.0039723661 0.0031088083 0.0017271157  
## [26] 0.0015544041 0.0001727116 0.0010362694 0.0010362694 0.0010362694  
## [31] 0.0003454231 0.0008635579 0.0005181347 0.0005181347 0.0008635579  
## [36] 0.0003454231 0.0001727116 0.0000000000 0.0003454231 0.0003454231  
## [41] 0.0005181347 0.0001727116 0.0010362694 0.0003454231 0.0003454231  
## [46] 0.0005181347 0.0001727116 0.0001727116 0.0001727116 0.0006908463  
## [51] 0.0001727116 0.0005181347 0.0001727116 0.0003454231 0.0001727116  
## [56] 0.0003454231 0.0003454231 0.0000000000 0.0001727116 0.0000000000  
## [61] 0.0005181347 0.0001727116 0.0003454231 0.0001727116 0.0000000000  
## [66] 0.0001727116 0.0000000000 0.0000000000 0.0005181347 0.0000000000  
## [71] 0.0001727116 0.0000000000 0.0000000000 0.0000000000 0.0003454231  
## [76] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0001727116  
## [81] 0.0001727116 0.0000000000 0.0000000000 0.0000000000 0.0001727116  
## [86] 0.0001727116 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [91] 0.0001727116 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [96] 0.0001727116 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [101] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0001727116  
## [106] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [111] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [116] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [121] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [126] 0.0000000000 0.0000000000 0.0000000000 0.0003454231 0.0000000000  
## [131] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [136] 0.0001727116 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [141] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [146] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [151] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [156] 0.0001727116 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [161] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [166] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [171] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [176] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [181] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [186] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [191] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [196] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [201] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0001727116  
##   
## $mids  
## [1] 65 75 85 95 105 115 125 135 145 155 165 175 185 195 205  
## [16] 215 225 235 245 255 265 275 285 295 305 315 325 335 345 355  
## [31] 365 375 385 395 405 415 425 435 445 455 465 475 485 495 505  
## [46] 515 525 535 545 555 565 575 585 595 605 615 625 635 645 655  
## [61] 665 675 685 695 705 715 725 735 745 755 765 775 785 795 805  
## [76] 815 825 835 845 855 865 875 885 895 905 915 925 935 945 955  
## [91] 965 975 985 995 1005 1015 1025 1035 1045 1055 1065 1075 1085 1095 1105  
## [106] 1115 1125 1135 1145 1155 1165 1175 1185 1195 1205 1215 1225 1235 1245 1255  
## [121] 1265 1275 1285 1295 1305 1315 1325 1335 1345 1355 1365 1375 1385 1395 1405  
## [136] 1415 1425 1435 1445 1455 1465 1475 1485 1495 1505 1515 1525 1535 1545 1555  
## [151] 1565 1575 1585 1595 1605 1615 1625 1635 1645 1655 1665 1675 1685 1695 1705  
## [166] 1715 1725 1735 1745 1755 1765 1775 1785 1795 1805 1815 1825 1835 1845 1855  
## [181] 1865 1875 1885 1895 1905 1915 1925 1935 1945 1955 1965 1975 1985 1995 2005  
## [196] 2015 2025 2035 2045 2055 2065 2075 2085 2095 2105  
##   
## $xname  
## [1] "liver$Alkaline\_Phosphotase"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Alkaline\_Phosphotase), max(liver$Alkaline\_Phosphotase),  
 length = length(liver$Alkaline\_Phosphotase)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Alkaline\_Phosphotase),  
 max(liver$Alkaline\_Phosphotase), length = length(liver$Alkaline\_Phosphotase)),  
 mu = mu.hat1, sigma = sigma.hat1), lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Alkaline\_Phosphotase), max(liver$Alkaline\_Phosphotase),  
 length = length(liver$Alkaline\_Phosphotase)),  
 mix.gam[["prob"]][2]\*dGA(seq(min(liver$Alkaline\_Phosphotase),  
 max(liver$Alkaline\_Phosphotase), length = length(liver$Alkaline\_Phosphotase)),  
 mu = mu.hat2, sigma = sigma.hat2), lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Alkaline\_Phosphotase), max(liver$Alkaline\_Phosphotase),  
 length = length(liver$Alkaline\_Phosphotase)),   
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Alkaline\_Phosphotase),  
 max(liver$Alkaline\_Phosphotase), length = length(liver$Alkaline\_Phosphotase)),  
 mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam[["prob"]][2]\*dRG(seq(min(liver$Alkaline\_Phosphotase),  
 max(liver$Alkaline\_Phosphotase), length = length(liver$Alkaline\_Phosphotase)),  
 mu = mu.hat2, sigma = sigma.hat2), lty = 3, lwd = 3, col = "green")

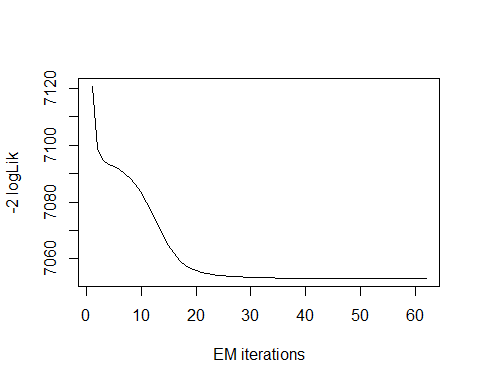


Now, I will again apply the fitting criteria for the distributions with Gamma mixture with the k = 3 as follows:

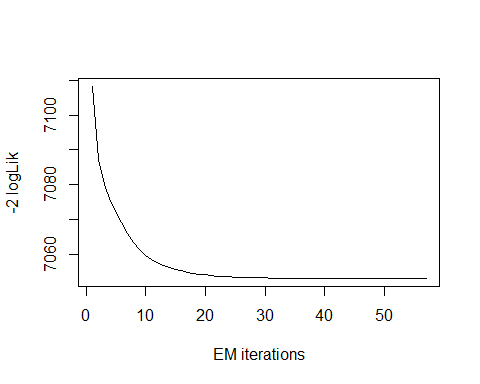
mix.gam.3 <- gamlssMXfits(n = 5, liver$Alkaline\_Phosphotase~1, family = GA, K = 3, data = liver)



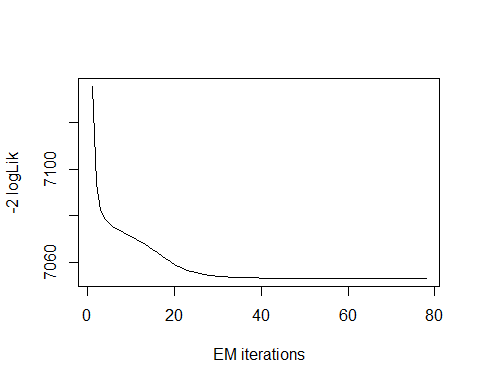
## model= 1



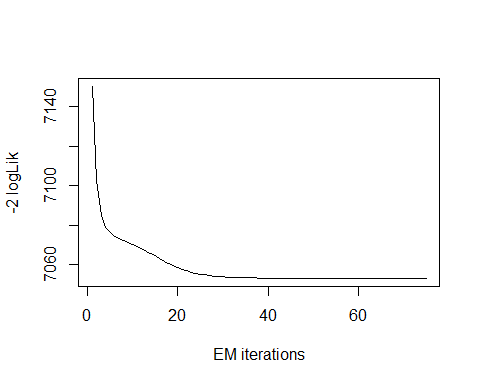
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam.3)

##   
## Mixing Family: c("GA", "GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Alkaline\_Phosphotase ~ 1,   
## family = GA, K = 3, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 5.477   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -1.057   
## Mu Coefficients for model: 2   
## (Intercept)   
## 6.424   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -0.4791   
## Mu Coefficients for model: 3   
## (Intercept)   
## 5.208   
## Sigma Coefficients for model: 3   
## (Intercept)   
## -2.079   
##   
## Estimated probabilities: 0.4881806 0.1869186 0.3249008   
##   
## Degrees of Freedom for the fit: 8 Residual Deg. of Freedom 571   
## Global Deviance: 7053.09   
## AIC: 7069.09   
## SBC: 7103.99

logLik(mix.gam.3)

## 'log Lik.' -3526.547 (df=8)

mix.gam.3$prob

## [1] 0.4881806 0.1869186 0.3249008

fitted(mix.gam.3, "mu")[1]

## [1] 291.3513

fitted(mix.gam.3, "sigma")[2]

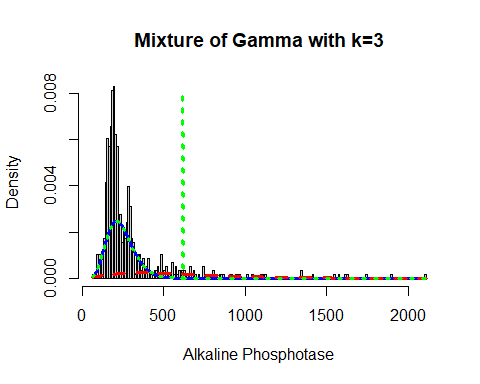
## [1] 291.3513

hist(liver$Alkaline\_Phosphotase, breaks = 262, xlab = "Alkaline Phosphotase", main="Mixture of Gamma with k=3", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam.3[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam.3[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam.3[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam.3[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Alkaline\_Phosphotase, breaks = 262, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Alkaline\_Phosphotase, breaks = 262, freq =  
## FALSE, : argument 'freq' is not made use of

## $breaks  
## [1] 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200  
## [16] 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350  
## [31] 360 370 380 390 400 410 420 430 440 450 460 470 480 490 500  
## [46] 510 520 530 540 550 560 570 580 590 600 610 620 630 640 650  
## [61] 660 670 680 690 700 710 720 730 740 750 760 770 780 790 800  
## [76] 810 820 830 840 850 860 870 880 890 900 910 920 930 940 950  
## [91] 960 970 980 990 1000 1010 1020 1030 1040 1050 1060 1070 1080 1090 1100  
## [106] 1110 1120 1130 1140 1150 1160 1170 1180 1190 1200 1210 1220 1230 1240 1250  
## [121] 1260 1270 1280 1290 1300 1310 1320 1330 1340 1350 1360 1370 1380 1390 1400  
## [136] 1410 1420 1430 1440 1450 1460 1470 1480 1490 1500 1510 1520 1530 1540 1550  
## [151] 1560 1570 1580 1590 1600 1610 1620 1630 1640 1650 1660 1670 1680 1690 1700  
## [166] 1710 1720 1730 1740 1750 1760 1770 1780 1790 1800 1810 1820 1830 1840 1850  
## [181] 1860 1870 1880 1890 1900 1910 1920 1930 1940 1950 1960 1970 1980 1990 2000  
## [196] 2010 2020 2030 2040 2050 2060 2070 2080 2090 2100 2110  
##   
## $counts  
## [1] 1 1 1 6 6 4 6 10 24 35 33 38 47 48 36 33 12 16 9 12 10 14 23 18 10  
## [26] 9 1 6 6 6 2 5 3 3 5 2 1 0 2 2 3 1 6 2 2 3 1 1 1 4  
## [51] 1 3 1 2 1 2 2 0 1 0 3 1 2 1 0 1 0 0 3 0 1 0 0 0 2  
## [76] 0 0 0 1 1 1 0 0 0 1 1 0 0 1 0 1 0 0 0 0 1 0 0 1 0  
## [101] 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [126] 0 0 0 2 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0  
## [151] 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0  
## [176] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [201] 0 0 0 0 1  
##   
## $density  
## [1] 0.0001727116 0.0001727116 0.0001727116 0.0010362694 0.0010362694  
## [6] 0.0006908463 0.0010362694 0.0017271157 0.0041450777 0.0060449050  
## [11] 0.0056994819 0.0065630397 0.0081174439 0.0082901554 0.0062176166  
## [16] 0.0056994819 0.0020725389 0.0027633851 0.0015544041 0.0020725389  
## [21] 0.0017271157 0.0024179620 0.0039723661 0.0031088083 0.0017271157  
## [26] 0.0015544041 0.0001727116 0.0010362694 0.0010362694 0.0010362694  
## [31] 0.0003454231 0.0008635579 0.0005181347 0.0005181347 0.0008635579  
## [36] 0.0003454231 0.0001727116 0.0000000000 0.0003454231 0.0003454231  
## [41] 0.0005181347 0.0001727116 0.0010362694 0.0003454231 0.0003454231  
## [46] 0.0005181347 0.0001727116 0.0001727116 0.0001727116 0.0006908463  
## [51] 0.0001727116 0.0005181347 0.0001727116 0.0003454231 0.0001727116  
## [56] 0.0003454231 0.0003454231 0.0000000000 0.0001727116 0.0000000000  
## [61] 0.0005181347 0.0001727116 0.0003454231 0.0001727116 0.0000000000  
## [66] 0.0001727116 0.0000000000 0.0000000000 0.0005181347 0.0000000000  
## [71] 0.0001727116 0.0000000000 0.0000000000 0.0000000000 0.0003454231  
## [76] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0001727116  
## [81] 0.0001727116 0.0000000000 0.0000000000 0.0000000000 0.0001727116  
## [86] 0.0001727116 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [91] 0.0001727116 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [96] 0.0001727116 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [101] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0001727116  
## [106] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [111] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [116] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [121] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [126] 0.0000000000 0.0000000000 0.0000000000 0.0003454231 0.0000000000  
## [131] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [136] 0.0001727116 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [141] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [146] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [151] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [156] 0.0001727116 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [161] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [166] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [171] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [176] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [181] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [186] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [191] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [196] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [201] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0001727116  
##   
## $mids  
## [1] 65 75 85 95 105 115 125 135 145 155 165 175 185 195 205  
## [16] 215 225 235 245 255 265 275 285 295 305 315 325 335 345 355  
## [31] 365 375 385 395 405 415 425 435 445 455 465 475 485 495 505  
## [46] 515 525 535 545 555 565 575 585 595 605 615 625 635 645 655  
## [61] 665 675 685 695 705 715 725 735 745 755 765 775 785 795 805  
## [76] 815 825 835 845 855 865 875 885 895 905 915 925 935 945 955  
## [91] 965 975 985 995 1005 1015 1025 1035 1045 1055 1065 1075 1085 1095 1105  
## [106] 1115 1125 1135 1145 1155 1165 1175 1185 1195 1205 1215 1225 1235 1245 1255  
## [121] 1265 1275 1285 1295 1305 1315 1325 1335 1345 1355 1365 1375 1385 1395 1405  
## [136] 1415 1425 1435 1445 1455 1465 1475 1485 1495 1505 1515 1525 1535 1545 1555  
## [151] 1565 1575 1585 1595 1605 1615 1625 1635 1645 1655 1665 1675 1685 1695 1705  
## [166] 1715 1725 1735 1745 1755 1765 1775 1785 1795 1805 1815 1825 1835 1845 1855  
## [181] 1865 1875 1885 1895 1905 1915 1925 1935 1945 1955 1965 1975 1985 1995 2005  
## [196] 2015 2025 2035 2045 2055 2065 2075 2085 2095 2105  
##   
## $xname  
## [1] "liver$Alkaline\_Phosphotase"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Alkaline\_Phosphotase), max(liver$Alkaline\_Phosphotase),  
 length = length(liver$Alkaline\_Phosphotase)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Alkaline\_Phosphotase),  
 max(liver$Alkaline\_Phosphotase), length = length(liver$Alkaline\_Phosphotase)),  
 mu = mu.hat1, sigma = sigma.hat1), lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Alkaline\_Phosphotase), max(liver$Alkaline\_Phosphotase),  
 length = length(liver$Alkaline\_Phosphotase)),  
 mix.gam.3[["prob"]][2]\*dGA(seq(min(liver$Alkaline\_Phosphotase),  
 max(liver$Alkaline\_Phosphotase), length = length(liver$Alkaline\_Phosphotase)),  
 mu = mu.hat2, sigma = sigma.hat2), lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Alkaline\_Phosphotase), max(liver$Alkaline\_Phosphotase),  
 length = length(liver$Alkaline\_Phosphotase)),   
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Alkaline\_Phosphotase),  
 max(liver$Alkaline\_Phosphotase), length = length(liver$Alkaline\_Phosphotase)),  
 mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam.3[["prob"]][2]\*dRG(seq(min(liver$Alkaline\_Phosphotase),  
 max(liver$Alkaline\_Phosphotase), length = length(liver$Alkaline\_Phosphotase)),  
 mu = mu.hat2, sigma = sigma.hat2), lty = 3, lwd = 3, col = "green")



mix.gm.tb <- data.frame(row.names=c('Gamma Mixture, K=3','Gamma Mixture, K=2', "Box-Cox Cole and Green Distribution"),  
 AIC=c(mix.gam.3$aic, mix.gam$aic, AIC(ap.BCCG)),  
 BIC=c(mix.gam.3$sbc, mix.gam$sbc, ap.BCCG$sbc))  
mix.gm.tb

## AIC BIC  
## Gamma Mixture, K=3 7069.095 7103.985  
## Gamma Mixture, K=2 7104.690 7126.497  
## Box-Cox Cole and Green Distribution 7132.341 7145.425

We can observe that the AIC value of the mixture of Gamma with k=3 and k=2 has improved, since values are lower than that of the Box-Cox Cole and Green Distribution and Gamma mixture with k=2. The previous AIC value was 7104.690, whereas the current value which is lower is 7069.095 and the previous BIC value was 7126.497, whereas the current value which is lower is 7103.986. Here we can clearly see that our data fits better in the Gamma Mixture Distribution with k=3.

## Alamine Aminotransferase

Lets explore the Alamine Aminotransferase variable:

head(liver$Alamine\_Aminotransferase)

## [1] 16 64 60 14 27 19

length(liver$Alamine\_Aminotransferase)

## [1] 579

table(liver$Alamine\_Aminotransferase)

##   
## 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25   
## 4 2 9 4 8 14 8 8 17 6 23 17 18 8 12 24   
## 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41   
## 10 9 17 12 15 12 12 10 3 9 11 9 8 4 9 5   
## 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57   
## 9 4 3 7 5 3 14 1 10 1 6 5 3 5 4 2   
## 58 59 60 61 62 63 64 65 67 68 69 70 71 72 74 75   
## 4 3 7 3 5 2 4 2 2 1 3 2 1 2 4 1   
## 76 78 79 80 82 84 85 86 88 89 90 91 93 94 95 96   
## 1 1 3 3 1 2 3 1 2 1 1 3 1 1 2 2   
## 97 99 102 107 110 112 114 115 116 118 119 120 123 126 131 132   
## 1 1 4 1 2 1 1 1 1 1 3 1 1 1 1 1   
## 133 137 139 140 141 142 148 149 152 154 155 157 159 160 166 168   
## 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2   
## 173 178 179 181 189 190 194 196 198 205 213 220 230 232 233 284   
## 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1   
## 308 321 322 349 378 382 390 404 407 412 425 440 482 509 622 779   
## 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1   
## 790 875 950 1250 1350 1630 1680 2000   
## 1 2 1 2 1 1 1 1

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.

unique(liver$Alamine\_Aminotransferase)

## [1] 16 64 60 14 27 19 22 53 51 31 61 91 168 15 232  
## [16] 17 116 52 875 1680 20 13 45 35 59 102 18 38 123 33  
## [31] 42 25 407 48 36 1630 39 21 80 86 26 24 37 40 62  
## [46] 55 166 189 95 12 194 58 28 119 412 404 220 126 190 97  
## [61] 308 32 29 11 63 181 88 74 2000 1350 1250 482 322 133 46  
## [76] 57 50 34 72 84 30 70 140 99 43 378 112 71 23 79  
## [91] 114 118 107 790 950 82 41 56 85 149 230 69 90 89 148  
## [106] 65 205 96 152 390 10 120 78 178 179 47 160 54 198 44  
## [121] 349 110 115 94 142 137 155 157 141 284 440 93 76 49 425  
## [136] 159 622 779 132 154 196 68 509 67 139 382 75 321 233 173  
## [151] 213 131

length(unique(liver$Alamine\_Aminotransferase))

## [1] 152

min(liver$Alamine\_Aminotransferase)

## [1] 10

max(liver$Alamine\_Aminotransferase)

## [1] 2000

Here the total observation of Age is 579 and is a continuous variable that range lies between 10-2000. Now, we will see the summary of the Alamine\_Aminotransferase variable, for further analysis as follows:

summary(liver$Alamine\_Aminotransferase)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 10.00 23.00 35.00 81.13 61.00 2000.00

sd(liver$Alamine\_Aminotransferase)

## [1] 183.1828

var(liver$Alamine\_Aminotransferase)

## [1] 33555.95

v <- c(liver$Alamine\_Aminotransferase)  
mode <- getMode(v)  
print(mode)

## [1] 25

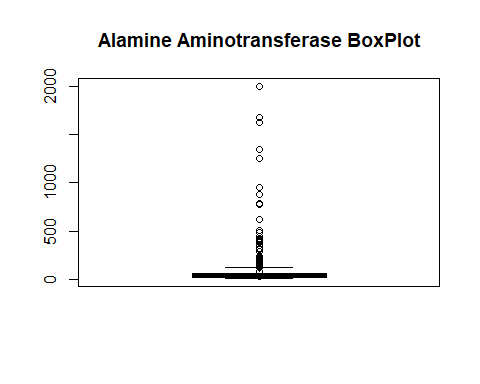
From the above summary we can observe that the Mean (81.13), Median (35.00) and Mode (25) are not equal so the distribution is asymmetrical. Here Mean > Median > Mode so the distribution could be positive and skewed to the right. Now we also confirm this as follows:

library(moments)  
skewness(liver$Alamine\_Aminotransferase)

## [1] 6.510652

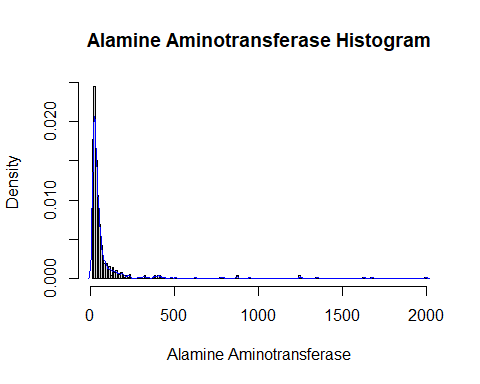
Hence, the skewness is the positive so the distribution is positively skewed was rightly observed. Also I will plot the Boxplot to comparing the distribution of data across data set as follows:

boxplot(liver$Alamine\_Aminotransferase, main = "Alamine Aminotransferase BoxPlot")  
points(mean(liver$Alamine\_Aminotransferase))  
points(mode)



Box plots are graphical displays of the five number summary, So they describe the Minimum (lowest mean relative) lower whisker, Quartile 1 (It separates the lowest 25% observation to the rest of the observations), Median (Which divides the lower 50% observation from the upper 50% observations), Quartile 3 (It divides the upper 25% observation to the rest of the observations) and Maximum. In the above Box plot diagram, there are many outliers and the upper whisker shows the highest mean relative. Now, I will plot a histogram a graphical display of data using bars of different heights to measures distribution of continuous data as follows:

hist(liver$Alamine\_Aminotransferase, prob = TRUE, breaks = 152, xlab = "Alamine Aminotransferase", main = "Alamine Aminotransferase Histogram")  
lines(density(liver$Alamine\_Aminotransferase), col='blue')



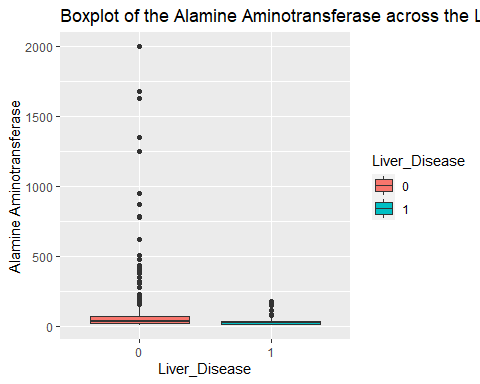
From the above histogram, we can observe that the diagram has many peaks.Using the density lines, I have approximated the histogram graph. Density provides information about the type of distribution under consideration and allows us to determine whether the attribute is distributed normally or not. From the graph above we can see how the Alamine\_Aminotransferase variable does not seem to fit to a Normal distribution, there is a peak of the frequency distributions around many values and also a right skewed. We have already observed a positive value of 6.510652, so we will affirm that the distribution is right skewed. To check whether the distribution is Platykurtic or not, we will check this by following R method as follows:

kurtosis(liver$Alamine\_Aminotransferase)

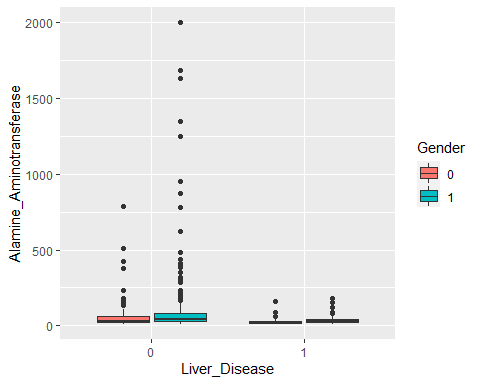
## [1] 52.79182

The distribution is also Leptokurtic, since the value is less than 3.

ggplot(liver, aes(x = Liver\_Disease, y = Alamine\_Aminotransferase, fill = Liver\_Disease)) +  
 geom\_boxplot() +  
 ylab("Alamine Aminotransferase") +  
 ggtitle("Boxplot of the Alamine Aminotransferase across the Liver\_Disease")



ggplot(liver, aes(Liver\_Disease, Alamine\_Aminotransferase)) +   
 geom\_boxplot(aes(fill = Gender))

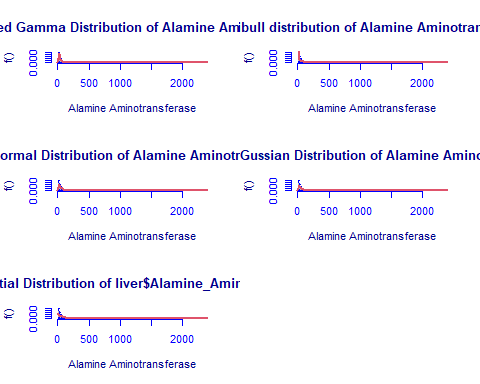


Another skewed variable. Range is from 10 to 2000, with a mean of 81.12, median of 35.00. There is a wider range within the Liver\_Disease = 0 group than the Liver\_Disease = 1 group, and the means are clearly different between the two groups. We again see a difference between the genders within the Liver\_Disease. The male groups within each group have a higher mean of Alamine Aminotransferase.

### Alamine Aminotransferase Fit for the Data

Now we will try to fit different models to Alamine Aminotransferase distribution evaluating the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC), The lower are the indexes, the better is the fitting. I will evaluate both the single parameter distributions and mixture distributions as follows:

library(MASS)  
library(gamlss)  
  
  
par(mfrow=c(3,2))  
  
aa.GG <- histDist(liver$Alamine\_Aminotransferase, family=GG, nbins = 152, xlab = "Alamine Aminotransferase", main="Generalized Gamma Distribution of Alamine Aminotransferase")  
  
aa.WEI <- histDist(liver$Alamine\_Aminotransferase, family=WEI, nbins = 152, xlab = "Alamine Aminotransferase", main="Weibull distribution of Alamine Aminotransferase")  
  
aa.LOGNO <- histDist(liver$Alamine\_Aminotransferase, family=LOGNO, nbins = 152, xlab = "Alamine Aminotransferase", main="Log-Normal Distribution of Alamine Aminotransferase")  
  
aa.IG <- histDist(liver$Alamine\_Aminotransferase, family=IG, nbins=152, xlab = "Alamine Aminotransferase", main = "Inverse Gussian Distribution of Alamine Aminotransferase")  
  
aa.EXP<- histDist(liver$Alamine\_Aminotransferase, family=EXP, nbins=152, xlab = "Alamine Aminotransferase", main = "Exponential Distribution of liver$Alamine\_Aminotransferase")



library(Matrix)  
library(glmnet)  
df <- data.frame(Rownames = c("Generalized Gamma", "Weibull", "Log-Normal",  
 "Inverse Gussian", "Exponential"),  
 AIC = c(AIC(aa.GG), AIC(aa.WEI), AIC(aa.LOGNO),  
 AIC(aa.IG), AIC(aa.EXP)),  
 BIC = c(aa.GG$sbc, aa.WEI$sbc, aa.LOGNO$sbc,  
 aa.IG$sbc, aa.EXP$sbc),  
 df = c(aa.GG$df.fit, aa.WEI$df.fit,  
 aa.LOGNO$df.fit, aa.IG$df.fit, aa.EXP$df.fit),  
 LogLike = c(logLik(aa.GG), logLik(aa.WEI),  
 logLik(aa.LOGNO), logLik(aa.IG), logLik(aa.EXP)))  
df

## Rownames AIC BIC df LogLike  
## 1 Generalized Gamma 5697.111 5710.195 3 -2845.555  
## 2 Weibull 6202.697 6211.420 2 -3099.349  
## 3 Log-Normal 5875.799 5884.522 2 -2935.900  
## 4 Inverse Gussian 5872.108 5880.830 2 -2934.054  
## 5 Exponential 6250.573 6254.934 1 -3124.287

As we can see, the model with the highest log likelihood (-2845.555) and the lowest AIC (5697.111) and BIC (5710.195) is the Generalized Gamma Distribution. Therefore, based on the greatest likelihood technique, our data fits better in the Generalized Gamma Distribution. Now, I will also compare two models by means of the Likelihood ratio test as we performed above as follows:

library(zoo)  
library(lmtest)  
lrtest(aa.GG, aa.EXP)

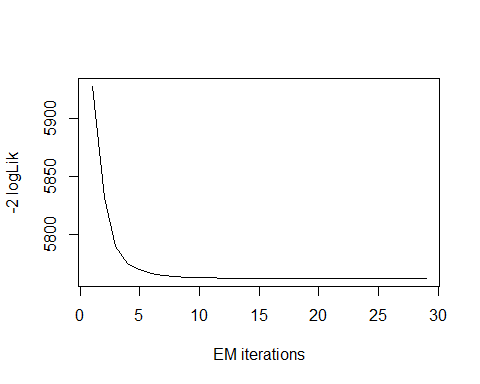
## Likelihood ratio test  
##   
## Model 1: gamlssML(formula = liver$Alamine\_Aminotransferase, family = "GG")  
## Model 2: gamlssML(formula = liver$Alamine\_Aminotransferase, family = "EXP")  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -2845.6   
## 2 1 -3124.3 -2 557.46 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Under the null hypothesis, we compare the Generalized Gamma distribution to the Exponential distribution under the alternative hypothesis. At any level of significance, we can reject the null hypothesis. Because our distributions fit better in the Generalized Gamma model, we’ll use it.

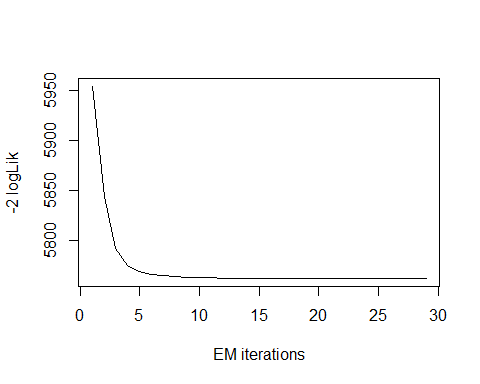
### Distributions Mixture

Now, I will also apply the fitting criteria for the distributions with Gamma mixture as follows:

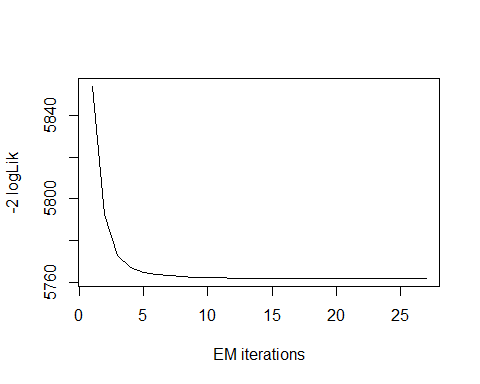
library(gamlss.mx)  
mix.gam <- gamlssMXfits(n = 5, liver$Alamine\_Aminotransferase~1, family = GA, K = 2, data = liver)



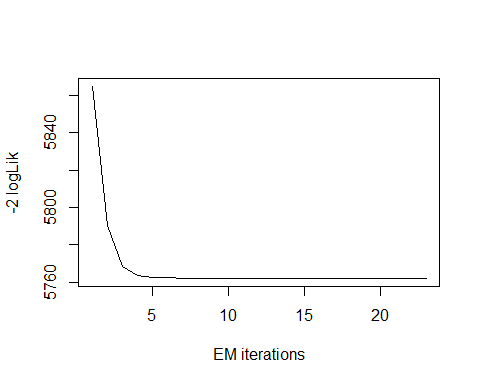
## model= 1



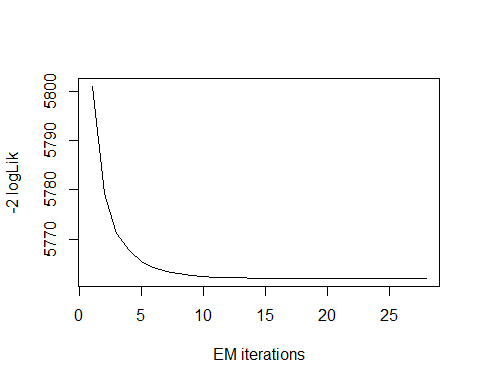
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam)

##   
## Mixing Family: c("GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Alamine\_Aminotransferase ~   
## 1, family = GA, K = 2, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 5.521   
## Sigma Coefficients for model: 1   
## (Intercept)   
## 0.005752   
## Mu Coefficients for model: 2   
## (Intercept)   
## 3.559   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -0.7189   
##   
## Estimated probabilities: 0.2141273 0.7858727   
##   
## Degrees of Freedom for the fit: 5 Residual Deg. of Freedom 574   
## Global Deviance: 5761.99   
## AIC: 5771.99   
## SBC: 5793.8

We can observe that the AIC value of the mixture of Gamma has improved, since it is higher than that of the single Gamma distribution. The current AIC value is 5771.99, whereas the previous value was 5697.111 and the current BIC value is 5793.8, whereas the previous value was 5710.195.

logLik(mix.gam)

## 'log Lik.' -2880.997 (df=5)

mix.gam$prob

## [1] 0.2141273 0.7858727

fitted(mix.gam, "mu")[1]

## [1] 81.11484

fitted(mix.gam, "sigma")[2]

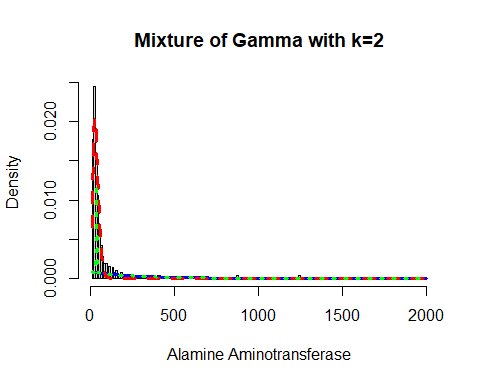
## [1] 81.11484

hist(liver$Alamine\_Aminotransferase, breaks = 152, xlab = "Alamine Aminotransferase", main="Mixture of Gamma with k=2", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Alamine\_Aminotransferase, breaks = 152, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Alamine\_Aminotransferase, breaks = 152, freq =  
## FALSE, : argument 'freq' is not made use of

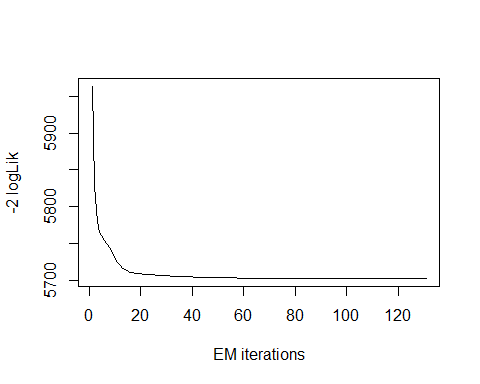
## $breaks  
## [1] 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150  
## [16] 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300  
## [31] 310 320 330 340 350 360 370 380 390 400 410 420 430 440 450  
## [46] 460 470 480 490 500 510 520 530 540 550 560 570 580 590 600  
## [61] 610 620 630 640 650 660 670 680 690 700 710 720 730 740 750  
## [76] 760 770 780 790 800 810 820 830 840 850 860 870 880 890 900  
## [91] 910 920 930 940 950 960 970 980 990 1000 1010 1020 1030 1040 1050  
## [106] 1060 1070 1080 1090 1100 1110 1120 1130 1140 1150 1160 1170 1180 1190 1200  
## [121] 1210 1220 1230 1240 1250 1260 1270 1280 1290 1300 1310 1320 1330 1340 1350  
## [136] 1360 1370 1380 1390 1400 1410 1420 1430 1440 1450 1460 1470 1480 1490 1500  
## [151] 1510 1520 1530 1540 1550 1560 1570 1580 1590 1600 1610 1620 1630 1640 1650  
## [166] 1660 1670 1680 1690 1700 1710 1720 1730 1740 1750 1760 1770 1780 1790 1800  
## [181] 1810 1820 1830 1840 1850 1860 1870 1880 1890 1900 1910 1920 1930 1940 1950  
## [196] 1960 1970 1980 1990 2000  
##   
## $counts  
## [1] 103 142 87 61 40 24 16 11 11 7 9 2 8 4 6 3 3 4  
## [19] 3 1 2 1 3 0 0 0 0 1 0 1 0 2 0 1 0 0  
## [37] 1 2 0 2 2 1 1 0 0 0 0 1 0 1 0 0 0 0  
## [55] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0  
## [73] 0 0 0 0 1 1 0 0 0 0 0 0 0 0 2 0 0 0  
## [91] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [109] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0  
## [127] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0  
## [145] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1  
## [163] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [181] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [199] 1  
##   
## $density  
## [1] 0.0177892919 0.0245250432 0.0150259067 0.0105354059 0.0069084629  
## [6] 0.0041450777 0.0027633851 0.0018998273 0.0018998273 0.0012089810  
## [11] 0.0015544041 0.0003454231 0.0013816926 0.0006908463 0.0010362694  
## [16] 0.0005181347 0.0005181347 0.0006908463 0.0005181347 0.0001727116  
## [21] 0.0003454231 0.0001727116 0.0005181347 0.0000000000 0.0000000000  
## [26] 0.0000000000 0.0000000000 0.0001727116 0.0000000000 0.0001727116  
## [31] 0.0000000000 0.0003454231 0.0000000000 0.0001727116 0.0000000000  
## [36] 0.0000000000 0.0001727116 0.0003454231 0.0000000000 0.0003454231  
## [41] 0.0003454231 0.0001727116 0.0001727116 0.0000000000 0.0000000000  
## [46] 0.0000000000 0.0000000000 0.0001727116 0.0000000000 0.0001727116  
## [51] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [56] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [61] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [66] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [71] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [76] 0.0000000000 0.0001727116 0.0001727116 0.0000000000 0.0000000000  
## [81] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [86] 0.0000000000 0.0003454231 0.0000000000 0.0000000000 0.0000000000  
## [91] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [96] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [101] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [106] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [111] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [116] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [121] 0.0000000000 0.0000000000 0.0000000000 0.0003454231 0.0000000000  
## [126] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [131] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [136] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [141] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [146] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [151] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [156] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [161] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [166] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [171] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [176] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [181] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [186] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [191] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [196] 0.0000000000 0.0000000000 0.0000000000 0.0001727116  
##   
## $mids  
## [1] 15 25 35 45 55 65 75 85 95 105 115 125 135 145 155  
## [16] 165 175 185 195 205 215 225 235 245 255 265 275 285 295 305  
## [31] 315 325 335 345 355 365 375 385 395 405 415 425 435 445 455  
## [46] 465 475 485 495 505 515 525 535 545 555 565 575 585 595 605  
## [61] 615 625 635 645 655 665 675 685 695 705 715 725 735 745 755  
## [76] 765 775 785 795 805 815 825 835 845 855 865 875 885 895 905  
## [91] 915 925 935 945 955 965 975 985 995 1005 1015 1025 1035 1045 1055  
## [106] 1065 1075 1085 1095 1105 1115 1125 1135 1145 1155 1165 1175 1185 1195 1205  
## [121] 1215 1225 1235 1245 1255 1265 1275 1285 1295 1305 1315 1325 1335 1345 1355  
## [136] 1365 1375 1385 1395 1405 1415 1425 1435 1445 1455 1465 1475 1485 1495 1505  
## [151] 1515 1525 1535 1545 1555 1565 1575 1585 1595 1605 1615 1625 1635 1645 1655  
## [166] 1665 1675 1685 1695 1705 1715 1725 1735 1745 1755 1765 1775 1785 1795 1805  
## [181] 1815 1825 1835 1845 1855 1865 1875 1885 1895 1905 1915 1925 1935 1945 1955  
## [196] 1965 1975 1985 1995  
##   
## $xname  
## [1] "liver$Alamine\_Aminotransferase"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Alamine\_Aminotransferase), max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Alamine\_Aminotransferase),  
 max(liver$Alamine\_Aminotransferase), length =length(liver$Alamine\_Aminotransferase)),  
 mu = mu.hat1, sigma = sigma.hat1), lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Alamine\_Aminotransferase), max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)),  
 mix.gam[["prob"]][2]\*dGA(seq(min(liver$Alamine\_Aminotransferase),  
 max(liver$Alamine\_Aminotransferase), length =length(liver$Alamine\_Aminotransferase)),  
 mu = mu.hat2, sigma = sigma.hat2), lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Alamine\_Aminotransferase), max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Alamine\_Aminotransferase),  
 max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)), mu = mu.hat1, sigma = sigma.hat1) +  
 mix.gam[["prob"]][2]\*dRG(seq(min(liver$Alamine\_Aminotransferase),  
 max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")

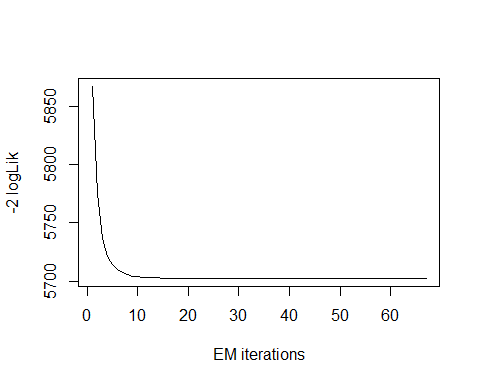


Now, I will again apply the fitting criteria for the distributions with Gamma mixture with the k = 3 as follows:

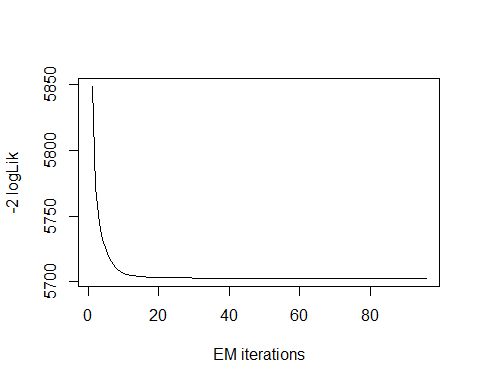
mix.gam.3 <- gamlssMXfits(n = 5, liver$Alamine\_Aminotransferase~1, family = GA, K = 3, data = liver)



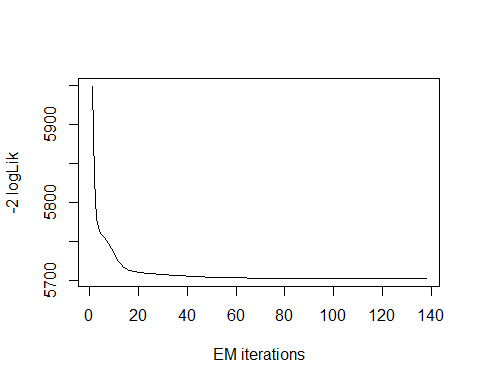
## model= 1



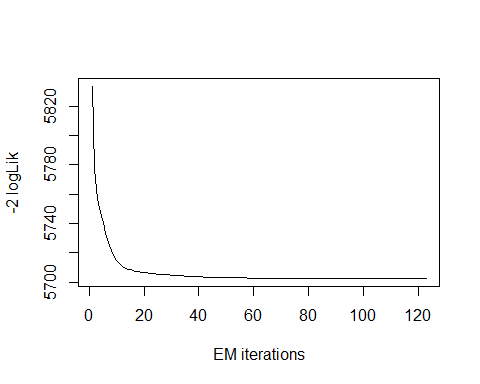
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam.3)

##   
## Mixing Family: c("GA", "GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Alamine\_Aminotransferase ~   
## 1, family = GA, K = 3, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 3.393   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -0.8918   
## Mu Coefficients for model: 2   
## (Intercept)   
## 6.256   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -0.1241   
## Mu Coefficients for model: 3   
## (Intercept)   
## 4.432   
## Sigma Coefficients for model: 3   
## (Intercept)   
## -0.5418   
##   
## Estimated probabilities: 0.6458668 0.07345744 0.2806758   
##   
## Degrees of Freedom for the fit: 8 Residual Deg. of Freedom 571   
## Global Deviance: 5702.46   
## AIC: 5718.46   
## SBC: 5753.35

logLik(mix.gam.3)

## 'log Lik.' -2851.231 (df=8)

mix.gam.3$prob

## [1] 0.64586680 0.07345744 0.28067576

fitted(mix.gam.3, "mu")[1]

## [1] 81.11845

fitted(mix.gam.3, "sigma")[2]

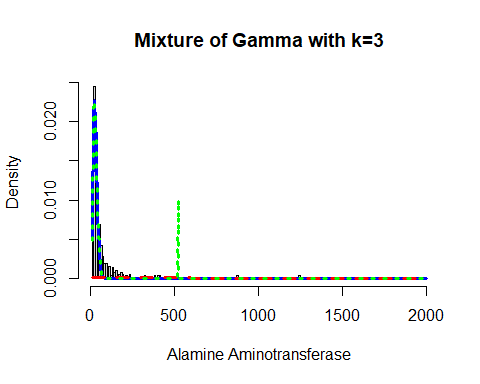
## [1] 81.11845

hist(liver$Alamine\_Aminotransferase, breaks = 152, xlab = "Alamine Aminotransferase", main="Mixture of Gamma with k=3", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam.3[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam.3[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam.3[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam.3[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Alamine\_Aminotransferase, breaks = 152, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Alamine\_Aminotransferase, breaks = 152, freq =  
## FALSE, : argument 'freq' is not made use of

## $breaks  
## [1] 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150  
## [16] 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300  
## [31] 310 320 330 340 350 360 370 380 390 400 410 420 430 440 450  
## [46] 460 470 480 490 500 510 520 530 540 550 560 570 580 590 600  
## [61] 610 620 630 640 650 660 670 680 690 700 710 720 730 740 750  
## [76] 760 770 780 790 800 810 820 830 840 850 860 870 880 890 900  
## [91] 910 920 930 940 950 960 970 980 990 1000 1010 1020 1030 1040 1050  
## [106] 1060 1070 1080 1090 1100 1110 1120 1130 1140 1150 1160 1170 1180 1190 1200  
## [121] 1210 1220 1230 1240 1250 1260 1270 1280 1290 1300 1310 1320 1330 1340 1350  
## [136] 1360 1370 1380 1390 1400 1410 1420 1430 1440 1450 1460 1470 1480 1490 1500  
## [151] 1510 1520 1530 1540 1550 1560 1570 1580 1590 1600 1610 1620 1630 1640 1650  
## [166] 1660 1670 1680 1690 1700 1710 1720 1730 1740 1750 1760 1770 1780 1790 1800  
## [181] 1810 1820 1830 1840 1850 1860 1870 1880 1890 1900 1910 1920 1930 1940 1950  
## [196] 1960 1970 1980 1990 2000  
##   
## $counts  
## [1] 103 142 87 61 40 24 16 11 11 7 9 2 8 4 6 3 3 4  
## [19] 3 1 2 1 3 0 0 0 0 1 0 1 0 2 0 1 0 0  
## [37] 1 2 0 2 2 1 1 0 0 0 0 1 0 1 0 0 0 0  
## [55] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0  
## [73] 0 0 0 0 1 1 0 0 0 0 0 0 0 0 2 0 0 0  
## [91] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [109] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0  
## [127] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0  
## [145] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1  
## [163] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [181] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [199] 1  
##   
## $density  
## [1] 0.0177892919 0.0245250432 0.0150259067 0.0105354059 0.0069084629  
## [6] 0.0041450777 0.0027633851 0.0018998273 0.0018998273 0.0012089810  
## [11] 0.0015544041 0.0003454231 0.0013816926 0.0006908463 0.0010362694  
## [16] 0.0005181347 0.0005181347 0.0006908463 0.0005181347 0.0001727116  
## [21] 0.0003454231 0.0001727116 0.0005181347 0.0000000000 0.0000000000  
## [26] 0.0000000000 0.0000000000 0.0001727116 0.0000000000 0.0001727116  
## [31] 0.0000000000 0.0003454231 0.0000000000 0.0001727116 0.0000000000  
## [36] 0.0000000000 0.0001727116 0.0003454231 0.0000000000 0.0003454231  
## [41] 0.0003454231 0.0001727116 0.0001727116 0.0000000000 0.0000000000  
## [46] 0.0000000000 0.0000000000 0.0001727116 0.0000000000 0.0001727116  
## [51] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [56] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [61] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [66] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [71] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [76] 0.0000000000 0.0001727116 0.0001727116 0.0000000000 0.0000000000  
## [81] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [86] 0.0000000000 0.0003454231 0.0000000000 0.0000000000 0.0000000000  
## [91] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [96] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [101] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [106] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [111] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [116] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [121] 0.0000000000 0.0000000000 0.0000000000 0.0003454231 0.0000000000  
## [126] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [131] 0.0000000000 0.0000000000 0.0000000000 0.0001727116 0.0000000000  
## [136] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [141] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [146] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [151] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [156] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [161] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [166] 0.0000000000 0.0001727116 0.0000000000 0.0000000000 0.0000000000  
## [171] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [176] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [181] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [186] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [191] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000  
## [196] 0.0000000000 0.0000000000 0.0000000000 0.0001727116  
##   
## $mids  
## [1] 15 25 35 45 55 65 75 85 95 105 115 125 135 145 155  
## [16] 165 175 185 195 205 215 225 235 245 255 265 275 285 295 305  
## [31] 315 325 335 345 355 365 375 385 395 405 415 425 435 445 455  
## [46] 465 475 485 495 505 515 525 535 545 555 565 575 585 595 605  
## [61] 615 625 635 645 655 665 675 685 695 705 715 725 735 745 755  
## [76] 765 775 785 795 805 815 825 835 845 855 865 875 885 895 905  
## [91] 915 925 935 945 955 965 975 985 995 1005 1015 1025 1035 1045 1055  
## [106] 1065 1075 1085 1095 1105 1115 1125 1135 1145 1155 1165 1175 1185 1195 1205  
## [121] 1215 1225 1235 1245 1255 1265 1275 1285 1295 1305 1315 1325 1335 1345 1355  
## [136] 1365 1375 1385 1395 1405 1415 1425 1435 1445 1455 1465 1475 1485 1495 1505  
## [151] 1515 1525 1535 1545 1555 1565 1575 1585 1595 1605 1615 1625 1635 1645 1655  
## [166] 1665 1675 1685 1695 1705 1715 1725 1735 1745 1755 1765 1775 1785 1795 1805  
## [181] 1815 1825 1835 1845 1855 1865 1875 1885 1895 1905 1915 1925 1935 1945 1955  
## [196] 1965 1975 1985 1995  
##   
## $xname  
## [1] "liver$Alamine\_Aminotransferase"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Alamine\_Aminotransferase), max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Alamine\_Aminotransferase),  
 max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Alamine\_Aminotransferase), max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)),  
 mix.gam.3[["prob"]][2]\*dGA(seq(min(liver$Alamine\_Aminotransferase),  
 max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Alamine\_Aminotransferase), max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Alamine\_Aminotransferase),  
 max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)),  
 mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam.3[["prob"]][2]\*dRG(seq(min(liver$Alamine\_Aminotransferase),  
 max(liver$Alamine\_Aminotransferase),  
 length = length(liver$Alamine\_Aminotransferase)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")



mix.gm.tb <- data.frame(row.names=c('Gamma Mixture, K=3','Gamma Mixture, K=2', "Generalized Gamma"),  
 AIC=c(mix.gam.3$aic, mix.gam$aic, AIC(aa.GG)),  
 BIC=c(mix.gam.3$sbc, mix.gam$sbc, aa.GG$sbc))  
mix.gm.tb

## AIC BIC  
## Gamma Mixture, K=3 5718.463 5753.353  
## Gamma Mixture, K=2 5771.995 5793.801  
## Generalized Gamma 5697.111 5710.195

We can observe that the AIC value of the mixture of Gamma with k=3 and k=2 has increased, since values are higher than that of the single Gamma distribution. The previous AIC value is 5697.111, whereas the current value which is higher is 5718.463 and the previous BIC value was 5710.195, whereas the current value which is higher is 5753.353. Here we can clearly see that our data fits better in the single Gamma Distribution.

## Aspartate Aminotransferase

Lets explore the Aspartate Aminotransferase variable:

head(liver$Aspartate\_Aminotransferase)

## [1] 18 100 68 20 59 14

length(liver$Aspartate\_Aminotransferase)

## [1] 579

table(liver$Aspartate\_Aminotransferase)

##   
## 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25   
## 1 2 5 3 8 10 9 8 9 11 14 14 13 16 12 13   
## 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41   
## 9 8 13 11 14 8 12 6 12 8 5 4 5 7 11 7   
## 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57   
## 9 7 6 4 3 5 5 1 4 4 2 5 6 2 5 6   
## 58 59 60 61 62 63 64 65 66 67 68 70 71 72 73 74   
## 9 4 1 2 3 2 1 3 6 2 5 3 2 1 3 2   
## 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90   
## 2 1 1 1 3 3 1 2 2 1 1 2 4 4 1 4   
## 91 92 95 97 98 99 100 101 102 103 104 105 108 110 111 113   
## 1 5 2 1 1 1 1 1 1 2 2 2 3 1 2 3   
## 114 116 125 126 127 130 134 135 138 139 140 141 142 143 145 148   
## 1 1 2 1 2 1 1 1 3 2 4 2 1 2 2 1   
## 149 150 152 155 156 161 168 176 178 180 181 185 186 187 188 190   
## 1 1 2 1 1 1 1 1 2 2 1 1 2 1 1 2   
## 200 202 220 221 230 231 232 233 235 236 245 247 248 250 275 285   
## 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1   
## 298 330 348 350 367 368 384 397 400 401 405 406 441 497 500 511   
## 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1   
## 540 562 576 602 623 630 731 794 844 850 950 960 1050 1500 1600 2946   
## 1 1 1 1 1 2 2 1 1 4 1 1 2 1 1 1   
## 4929   
## 1

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.

unique(liver$Aspartate\_Aminotransferase)

## [1] 18 100 68 20 59 14 12 11 19 58 56 30 41 53 441  
## [16] 23 245 28 34 66 55 45 731 850 21 111 44 57 80 36  
## [31] 77 73 50 110 47 576 15 178 27 960 406 150 61 54 24  
## [46] 16 43 97 86 88 95 26 17 397 29 22 127 79 142 152  
## [61] 31 350 794 400 202 630 950 161 405 92 39 10 116 98 285  
## [76] 64 149 2946 1600 1050 275 113 84 25 40 83 65 4929 90 140  
## [91] 139 87 38 42 233 138 82 35 32 187 62 74 67 37 602  
## [106] 63 99 103 145 247 114 104 51 60 1500 180 148 46 13 85  
## [121] 231 156 89 298 48 130 75 500 105 250 232 33 143 176 70  
## [136] 52 91 236 108 190 71 126 141 102 81 511 72 135 497 844  
## [151] 368 188 248 401 76 221 235 185 230 540 181 155 200 186 623  
## [166] 220 78 348 125 330 562 384 367 101 168 134 49

length(unique(liver$Aspartate\_Aminotransferase))

## [1] 177

min(liver$Aspartate\_Aminotransferase)

## [1] 10

max(liver$Aspartate\_Aminotransferase)

## [1] 4929

Here the total observation of Aspartate Aminotransferase is 579 and is a continuous variable that range lies between 10-4929. Now, we will see the summary of the Aspartate\_Aminotransferase variable, for further analysis as follows:

summary(liver$Aspartate\_Aminotransferase)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 10.0 25.0 42.0 110.4 87.0 4929.0

sd(liver$Aspartate\_Aminotransferase)

## [1] 289.85

var(liver$Aspartate\_Aminotransferase)

## [1] 84013.04

v <- c(liver$Aspartate\_Aminotransferase)  
mode <- getMode(v)  
print(mode)

## [1] 23

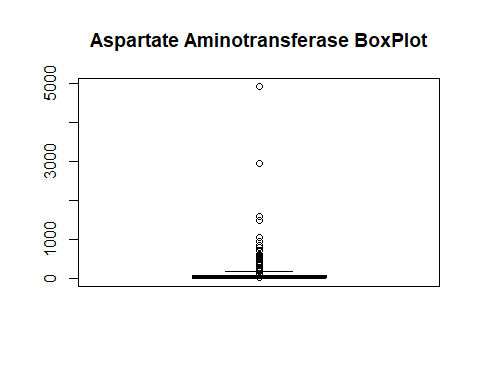
From the above summary we can observe that the Mean (110.4), Median (42.0) and Mode (23) are not equal so the distribution is asymmetrical. Here Mean > Median > Mode so the distribution could be positive and skewed to the right. Now we also confirm this as follows:

library(moments)  
skewness(liver$Aspartate\_Aminotransferase)

## [1] 10.485

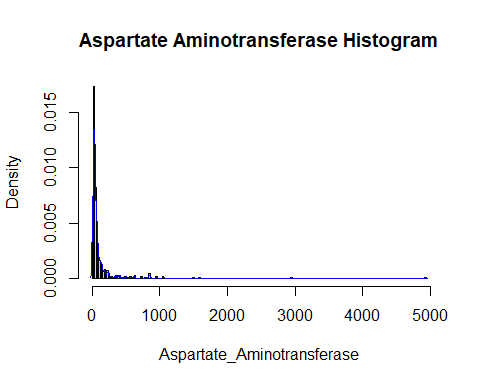
Hence, the skewness is the positive so the distribution is positively skewed was rightly observed. Also I will plot the Boxplot to comparing the distribution of data across data set as follows:

boxplot(liver$Aspartate\_Aminotransferase, main = "Aspartate Aminotransferase BoxPlot")  
points(mean(liver$Aspartate\_Aminotransferase))  
points(mode)



Box plots are graphical displays of the five number summary, So they describe the Minimum (lowest mean relative) lower whisker, Quartile 1 (It separates the lowest 25% observation to the rest of the observations), Median (Which divides the lower 50% observation from the upper 50% observations), Quartile 3 (It divides the upper 25% observation to the rest of the observations) and Maximum. In the above Box plot diagram, there are many outliers and the upper whisker shows the highest mean relative. Now, I will plot a histogram a graphical display of data using bars of different heights to measures distribution of continuous data as follows:

hist(liver$Aspartate\_Aminotransferase, prob = TRUE, breaks = 177, xlab = "Aspartate\_Aminotransferase", main = "Aspartate Aminotransferase Histogram")  
lines(density(liver$Aspartate\_Aminotransferase), col='blue')



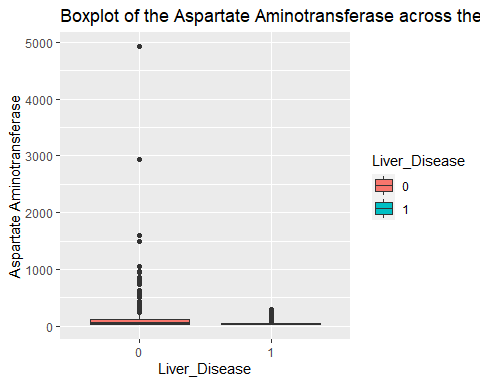
From the above histogram, we can observe that the diagram has many peaks.Using the density lines, I have approximated the histogram graph. Density provides information about the type of distribution under consideration and allows us to determine whether the attribute is distributed normally or not. From the graph above we can see how the Aspartate\_Aminotransferase variable does not seem to fit to a Normal distribution, there is a peak of the frequency distributions around many values and also a right skewed. We have already observed a positive value of 10.485, so we will affirm that the distribution is right skewed. To check whether the distribution is Platykurtic or not, we will check this by following R method as follows:

kurtosis(liver$Aspartate\_Aminotransferase)

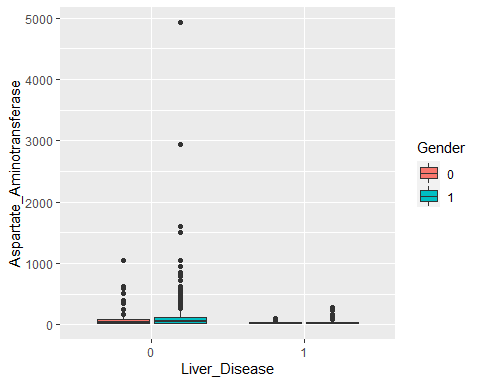
## [1] 151.6374

The distribution is also Leptokurtic, since the value is greater than 3.

ggplot(liver, aes(x = Liver\_Disease, y = Aspartate\_Aminotransferase,  
 fill = Liver\_Disease)) +  
 geom\_boxplot() +  
 ylab("Aspartate Aminotransferase") +  
 ggtitle("Boxplot of the Aspartate Aminotransferase across the Liver\_Disease")



ggplot(liver, aes(Liver\_Disease, Aspartate\_Aminotransferase)) +   
 geom\_boxplot(aes(fill = Gender))

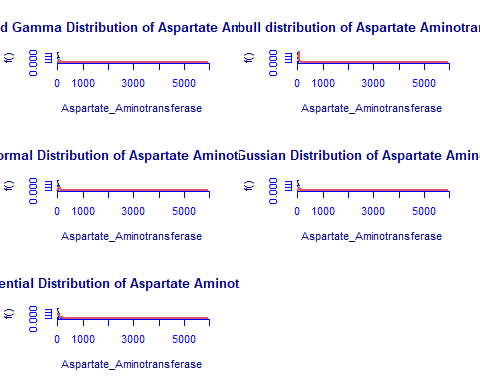


Another skewed variable with a range from 10 to 4929, with a mean of 110.4145, and median of 42. The range is wider within the Liver\_Disease = 0 group in comparison to the Liver\_Disease = 1 group. The mean is also higher in the Liver\_Disease = 0 group. The box plots are hard to really read but they are there for you to look at.

### Aspartate Aminotransferase Fit for the Data

Now we will try to fit different models to Aspartate\_Aminotransferase distribution evaluating the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC), The lower are the indexes, the better is the fitting. I will evaluate both the single parameter distributions and mixture distributions as follows:

library(MASS)  
library(gamlss)  
  
  
par(mfrow=c(3,2))  
  
aam.GG <- histDist(liver$Aspartate\_Aminotransferase, family=GG, nbins = 177, xlab = "Aspartate\_Aminotransferase", main="Generalized Gamma Distribution of Aspartate Aminotransferase")  
  
aam.WEI <- histDist(liver$Aspartate\_Aminotransferase, family=WEI, nbins = 177, xlab = "Aspartate\_Aminotransferase", main="Weibull distribution of Aspartate Aminotransferase")  
  
aam.LOGNO <- histDist(liver$Aspartate\_Aminotransferase, family=LOGNO, nbins = 177, xlab = "Aspartate\_Aminotransferase", main="Log-Normal Distribution of Aspartate Aminotransferase")  
  
aam.IG <- histDist(liver$Aspartate\_Aminotransferase, family=IG, nbins=177, xlab = "Aspartate\_Aminotransferase", main = "Inverse Gussian Distribution of Aspartate Aminotransferase")  
  
aam.EXP<- histDist(liver$Aspartate\_Aminotransferase, family=EXP, nbins=177, xlab = "Aspartate\_Aminotransferase", main = "Exponential Distribution of Aspartate Aminotransferase")



library(Matrix)  
library(glmnet)  
df <- data.frame(Rownames = c("Generalized Gamma", "Weibull",  
 "Log-Normal", "Inverse Gussian", "Exponential"),  
 AIC = c(AIC(aam.GG), AIC(aam.WEI), AIC(aam.LOGNO),  
 AIC(aam.IG), AIC(aam.EXP)),  
 BIC = c(aam.GG$sbc, aam.WEI$sbc, aam.LOGNO$sbc,  
 aam.IG$sbc, aam.EXP$sbc),  
 df = c(aam.GG$df.fit, aam.WEI$df.fit,  
 aam.LOGNO$df.fit, aam.IG$df.fit, aam.EXP$df.fit),  
 LogLike = c(logLik(aam.GG), logLik(aam.WEI),  
 logLik(aam.LOGNO), logLik(aam.IG), logLik(aam.EXP)))  
df

## Rownames AIC BIC df LogLike  
## 1 Generalized Gamma 6072.943 6086.027 3 -3033.472  
## 2 Weibull 6518.690 6527.413 2 -3257.345  
## 3 Log-Normal 6230.840 6239.562 2 -3113.420  
## 4 Inverse Gussian 6204.105 6212.827 2 -3100.052  
## 5 Exponential 6607.512 6611.873 1 -3302.756

As we can see, the model with the highest log likelihood (-3033.472) and the lowest AIC (6072.943) and BIC (6086.027) is the Generalized Gamma Distribution. Therefore, based on the greatest likelihood technique, our data fits better in the Generalized Gamma Distribution. Now, I will also compare two models by means of the Likelihood ratio test as we performed above as follows:

library(zoo)  
library(lmtest)  
lrtest(aam.GG, aam.EXP)

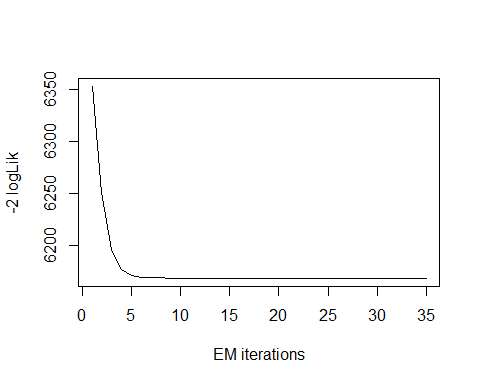
## Likelihood ratio test  
##   
## Model 1: gamlssML(formula = liver$Aspartate\_Aminotransferase, family = "GG")  
## Model 2: gamlssML(formula = liver$Aspartate\_Aminotransferase, family = "EXP")  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -3033.5   
## 2 1 -3302.8 -2 538.57 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Under the null hypothesis, we compare the Generalized Gamma distribution to the Exponential distribution under the alternative hypothesis. At any level of significance, we can reject the null hypothesis. Because our distributions fit better in the Generalized Gamma model, we’ll use it.

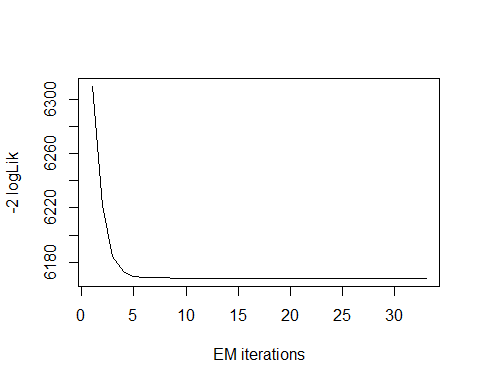
### Distributions Mixture

Now, I will also apply the fitting criteria for the distributions with Gamma mixture as follows:

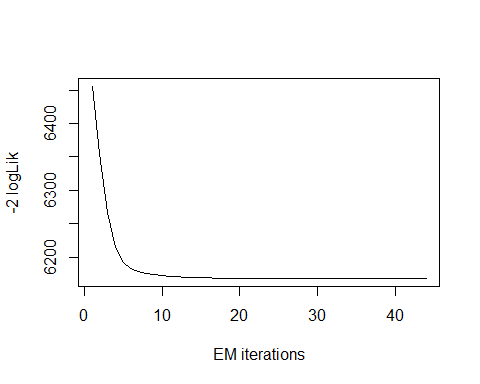
library(gamlss.mx)  
mix.gam <- gamlssMXfits(n = 5, liver$Aspartate\_Aminotransferase~1, family = GA, K = 2, data = liver)



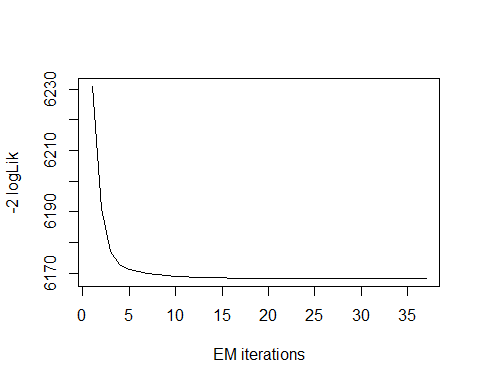
## model= 1



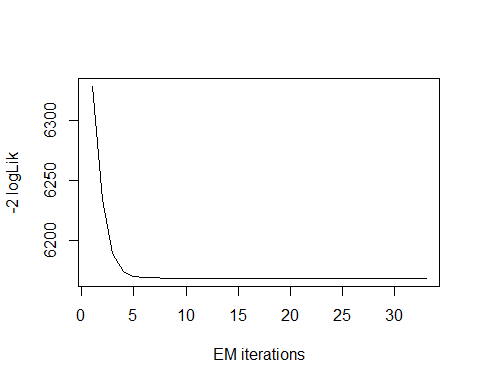
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam)

##   
## Mixing Family: c("GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Aspartate\_Aminotransferase ~   
## 1, family = GA, K = 2, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 5.737   
## Sigma Coefficients for model: 1   
## (Intercept)   
## 0.03021   
## Mu Coefficients for model: 2   
## (Intercept)   
## 3.712   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -0.609   
##   
## Estimated probabilities: 0.2580254 0.7419746   
##   
## Degrees of Freedom for the fit: 5 Residual Deg. of Freedom 574   
## Global Deviance: 6168.29   
## AIC: 6178.29   
## SBC: 6200.1

We can observe that the AIC value of the mixture of Gamma has improved, since it is higher than that of the single Gamma distribution. The current AIC value is 6178.29, whereas the previous value was 6072.943 and the current BIC value is 6200.1, whereas the previous value was 6086.027.

logLik(mix.gam)

## 'log Lik.' -3084.147 (df=5)

mix.gam$prob

## [1] 0.2580254 0.7419746

fitted(mix.gam, "mu")[1]

## [1] 110.3985

fitted(mix.gam, "sigma")[2]

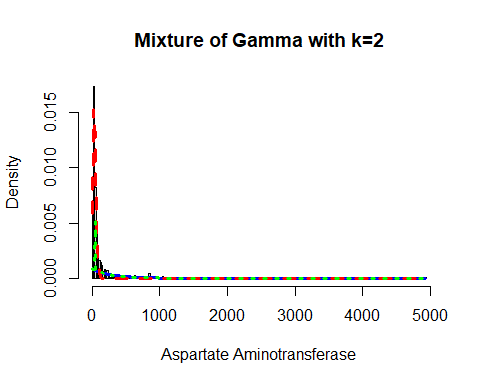
## [1] 110.3985

hist(liver$Aspartate\_Aminotransferase, breaks = 177, xlab = "Aspartate Aminotransferase", main="Mixture of Gamma with k=2", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Aspartate\_Aminotransferase, breaks = 177, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Aspartate\_Aminotransferase, breaks = 177, :  
## argument 'freq' is not made use of

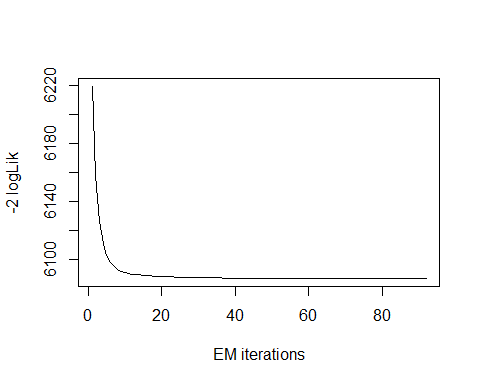
## $breaks  
## [1] 0 20 40 60 80 100 120 140 160 180 200 220 240 260 280  
## [16] 300 320 340 360 380 400 420 440 460 480 500 520 540 560 580  
## [31] 600 620 640 660 680 700 720 740 760 780 800 820 840 860 880  
## [46] 900 920 940 960 980 1000 1020 1040 1060 1080 1100 1120 1140 1160 1180  
## [61] 1200 1220 1240 1260 1280 1300 1320 1340 1360 1380 1400 1420 1440 1460 1480  
## [76] 1500 1520 1540 1560 1580 1600 1620 1640 1660 1680 1700 1720 1740 1760 1780  
## [91] 1800 1820 1840 1860 1880 1900 1920 1940 1960 1980 2000 2020 2040 2060 2080  
## [106] 2100 2120 2140 2160 2180 2200 2220 2240 2260 2280 2300 2320 2340 2360 2380  
## [121] 2400 2420 2440 2460 2480 2500 2520 2540 2560 2580 2600 2620 2640 2660 2680  
## [136] 2700 2720 2740 2760 2780 2800 2820 2840 2860 2880 2900 2920 2940 2960 2980  
## [151] 3000 3020 3040 3060 3080 3100 3120 3140 3160 3180 3200 3220 3240 3260 3280  
## [166] 3300 3320 3340 3360 3380 3400 3420 3440 3460 3480 3500 3520 3540 3560 3580  
## [181] 3600 3620 3640 3660 3680 3700 3720 3740 3760 3780 3800 3820 3840 3860 3880  
## [196] 3900 3920 3940 3960 3980 4000 4020 4040 4060 4080 4100 4120 4140 4160 4180  
## [211] 4200 4220 4240 4260 4280 4300 4320 4340 4360 4380 4400 4420 4440 4460 4480  
## [226] 4500 4520 4540 4560 4580 4600 4620 4640 4660 4680 4700 4720 4740 4760 4780  
## [241] 4800 4820 4840 4860 4880 4900 4920 4940  
##   
## $counts  
## [1] 80 201 95 46 34 19 17 14 7 9 2 8 5 1 2 0 1 3  
## [19] 2 3 3 0 1 0 2 1 1 0 2 0 1 3 0 0 0 0  
## [37] 2 0 0 1 0 0 5 0 0 0 0 2 0 0 0 0 2 0  
## [55] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [73] 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0  
## [91] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [109] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [127] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [145] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [163] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [181] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [199] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [217] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [235] 0 0 0 0 0 0 0 0 0 0 0 0 1  
##   
## $density  
## [1] 6.908463e-03 1.735751e-02 8.203800e-03 3.972366e-03 2.936097e-03  
## [6] 1.640760e-03 1.468048e-03 1.208981e-03 6.044905e-04 7.772021e-04  
## [11] 1.727116e-04 6.908463e-04 4.317789e-04 8.635579e-05 1.727116e-04  
## [16] 0.000000e+00 8.635579e-05 2.590674e-04 1.727116e-04 2.590674e-04  
## [21] 2.590674e-04 0.000000e+00 8.635579e-05 0.000000e+00 1.727116e-04  
## [26] 8.635579e-05 8.635579e-05 0.000000e+00 1.727116e-04 0.000000e+00  
## [31] 8.635579e-05 2.590674e-04 0.000000e+00 0.000000e+00 0.000000e+00  
## [36] 0.000000e+00 1.727116e-04 0.000000e+00 0.000000e+00 8.635579e-05  
## [41] 0.000000e+00 0.000000e+00 4.317789e-04 0.000000e+00 0.000000e+00  
## [46] 0.000000e+00 0.000000e+00 1.727116e-04 0.000000e+00 0.000000e+00  
## [51] 0.000000e+00 0.000000e+00 1.727116e-04 0.000000e+00 0.000000e+00  
## [56] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [61] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [66] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [71] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 8.635579e-05  
## [76] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 8.635579e-05  
## [81] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [86] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [91] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [96] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [101] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [106] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [111] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [116] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [121] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [126] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [131] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [136] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [141] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [146] 0.000000e+00 0.000000e+00 8.635579e-05 0.000000e+00 0.000000e+00  
## [151] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [156] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [161] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [166] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [171] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [176] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [181] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [186] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [191] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [196] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [201] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [206] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [211] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [216] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [221] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [226] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [231] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [236] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [241] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [246] 0.000000e+00 8.635579e-05  
##   
## $mids  
## [1] 10 30 50 70 90 110 130 150 170 190 210 230 250 270 290  
## [16] 310 330 350 370 390 410 430 450 470 490 510 530 550 570 590  
## [31] 610 630 650 670 690 710 730 750 770 790 810 830 850 870 890  
## [46] 910 930 950 970 990 1010 1030 1050 1070 1090 1110 1130 1150 1170 1190  
## [61] 1210 1230 1250 1270 1290 1310 1330 1350 1370 1390 1410 1430 1450 1470 1490  
## [76] 1510 1530 1550 1570 1590 1610 1630 1650 1670 1690 1710 1730 1750 1770 1790  
## [91] 1810 1830 1850 1870 1890 1910 1930 1950 1970 1990 2010 2030 2050 2070 2090  
## [106] 2110 2130 2150 2170 2190 2210 2230 2250 2270 2290 2310 2330 2350 2370 2390  
## [121] 2410 2430 2450 2470 2490 2510 2530 2550 2570 2590 2610 2630 2650 2670 2690  
## [136] 2710 2730 2750 2770 2790 2810 2830 2850 2870 2890 2910 2930 2950 2970 2990  
## [151] 3010 3030 3050 3070 3090 3110 3130 3150 3170 3190 3210 3230 3250 3270 3290  
## [166] 3310 3330 3350 3370 3390 3410 3430 3450 3470 3490 3510 3530 3550 3570 3590  
## [181] 3610 3630 3650 3670 3690 3710 3730 3750 3770 3790 3810 3830 3850 3870 3890  
## [196] 3910 3930 3950 3970 3990 4010 4030 4050 4070 4090 4110 4130 4150 4170 4190  
## [211] 4210 4230 4250 4270 4290 4310 4330 4350 4370 4390 4410 4430 4450 4470 4490  
## [226] 4510 4530 4550 4570 4590 4610 4630 4650 4670 4690 4710 4730 4750 4770 4790  
## [241] 4810 4830 4850 4870 4890 4910 4930  
##   
## $xname  
## [1] "liver$Aspartate\_Aminotransferase"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase), length = length(liver$Aspartate\_Aminotransferase)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase),  
 length = length(liver$Aspartate\_Aminotransferase)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase), length = length(liver$Aspartate\_Aminotransferase)),  
 mix.gam[["prob"]][2]\*dGA(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase),  
 length = length(liver$Aspartate\_Aminotransferase)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase), length = length(liver$Aspartate\_Aminotransferase)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase),  
 length = length(liver$Aspartate\_Aminotransferase)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam[["prob"]][2]\*dRG(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase),  
 length = length(liver$Aspartate\_Aminotransferase)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")

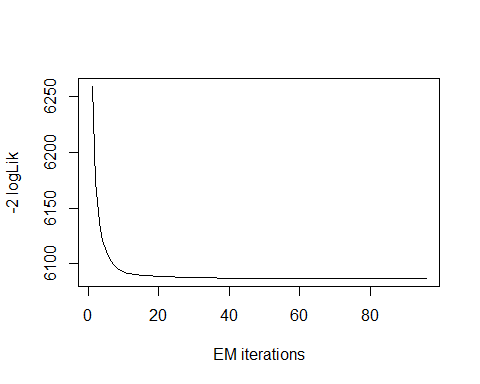


Now, I will again apply the fitting criteria for the distributions with Gamma mixture with the k = 3 as follows:

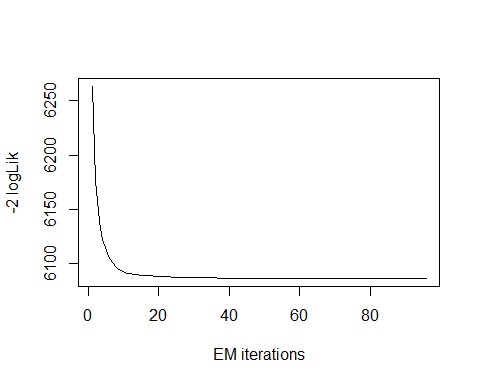
mix.gam.3 <- gamlssMXfits(n = 5, liver$Aspartate\_Aminotransferase~1, family = GA, K = 3, data = liver)



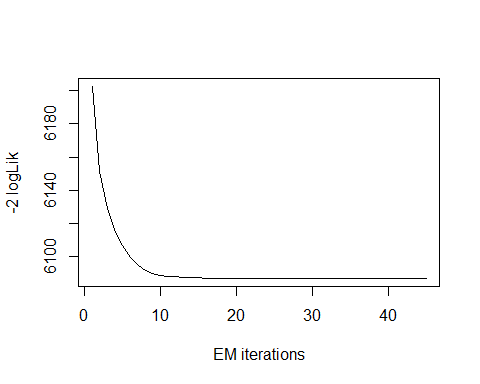
## model= 1



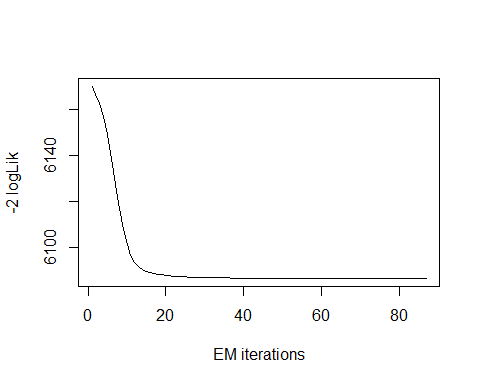
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam.3)

##   
## Mixing Family: c("GA", "GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Aspartate\_Aminotransferase ~   
## 1, family = GA, K = 3, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 3.347   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -0.9957   
## Mu Coefficients for model: 2   
## (Intercept)   
## 4.41   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -0.5401   
## Mu Coefficients for model: 3   
## (Intercept)   
## 6.256   
## Sigma Coefficients for model: 3   
## (Intercept)   
## -0.03811   
##   
## Estimated probabilities: 0.4764296 0.4009889 0.1225815   
##   
## Degrees of Freedom for the fit: 8 Residual Deg. of Freedom 571   
## Global Deviance: 6086.75   
## AIC: 6102.75   
## SBC: 6137.64

logLik(mix.gam.3)

## 'log Lik.' -3043.376 (df=8)

mix.gam.3$prob

## [1] 0.4764296 0.4009889 0.1225815

fitted(mix.gam.3, "mu")[1]

## [1] 110.3993

fitted(mix.gam.3, "sigma")[2]

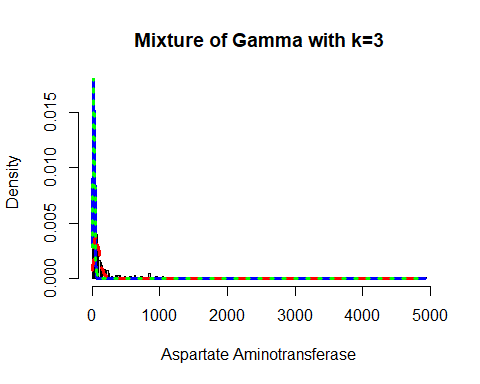
## [1] 110.3993

hist(liver$Aspartate\_Aminotransferase, breaks = 177, xlab = "Aspartate Aminotransferase", main="Mixture of Gamma with k=3", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam.3[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam.3[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam.3[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam.3[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Aspartate\_Aminotransferase, breaks = 177, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Aspartate\_Aminotransferase, breaks = 177, :  
## argument 'freq' is not made use of

## $breaks  
## [1] 0 20 40 60 80 100 120 140 160 180 200 220 240 260 280  
## [16] 300 320 340 360 380 400 420 440 460 480 500 520 540 560 580  
## [31] 600 620 640 660 680 700 720 740 760 780 800 820 840 860 880  
## [46] 900 920 940 960 980 1000 1020 1040 1060 1080 1100 1120 1140 1160 1180  
## [61] 1200 1220 1240 1260 1280 1300 1320 1340 1360 1380 1400 1420 1440 1460 1480  
## [76] 1500 1520 1540 1560 1580 1600 1620 1640 1660 1680 1700 1720 1740 1760 1780  
## [91] 1800 1820 1840 1860 1880 1900 1920 1940 1960 1980 2000 2020 2040 2060 2080  
## [106] 2100 2120 2140 2160 2180 2200 2220 2240 2260 2280 2300 2320 2340 2360 2380  
## [121] 2400 2420 2440 2460 2480 2500 2520 2540 2560 2580 2600 2620 2640 2660 2680  
## [136] 2700 2720 2740 2760 2780 2800 2820 2840 2860 2880 2900 2920 2940 2960 2980  
## [151] 3000 3020 3040 3060 3080 3100 3120 3140 3160 3180 3200 3220 3240 3260 3280  
## [166] 3300 3320 3340 3360 3380 3400 3420 3440 3460 3480 3500 3520 3540 3560 3580  
## [181] 3600 3620 3640 3660 3680 3700 3720 3740 3760 3780 3800 3820 3840 3860 3880  
## [196] 3900 3920 3940 3960 3980 4000 4020 4040 4060 4080 4100 4120 4140 4160 4180  
## [211] 4200 4220 4240 4260 4280 4300 4320 4340 4360 4380 4400 4420 4440 4460 4480  
## [226] 4500 4520 4540 4560 4580 4600 4620 4640 4660 4680 4700 4720 4740 4760 4780  
## [241] 4800 4820 4840 4860 4880 4900 4920 4940  
##   
## $counts  
## [1] 80 201 95 46 34 19 17 14 7 9 2 8 5 1 2 0 1 3  
## [19] 2 3 3 0 1 0 2 1 1 0 2 0 1 3 0 0 0 0  
## [37] 2 0 0 1 0 0 5 0 0 0 0 2 0 0 0 0 2 0  
## [55] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [73] 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0  
## [91] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [109] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [127] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [145] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [163] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [181] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [199] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [217] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [235] 0 0 0 0 0 0 0 0 0 0 0 0 1  
##   
## $density  
## [1] 6.908463e-03 1.735751e-02 8.203800e-03 3.972366e-03 2.936097e-03  
## [6] 1.640760e-03 1.468048e-03 1.208981e-03 6.044905e-04 7.772021e-04  
## [11] 1.727116e-04 6.908463e-04 4.317789e-04 8.635579e-05 1.727116e-04  
## [16] 0.000000e+00 8.635579e-05 2.590674e-04 1.727116e-04 2.590674e-04  
## [21] 2.590674e-04 0.000000e+00 8.635579e-05 0.000000e+00 1.727116e-04  
## [26] 8.635579e-05 8.635579e-05 0.000000e+00 1.727116e-04 0.000000e+00  
## [31] 8.635579e-05 2.590674e-04 0.000000e+00 0.000000e+00 0.000000e+00  
## [36] 0.000000e+00 1.727116e-04 0.000000e+00 0.000000e+00 8.635579e-05  
## [41] 0.000000e+00 0.000000e+00 4.317789e-04 0.000000e+00 0.000000e+00  
## [46] 0.000000e+00 0.000000e+00 1.727116e-04 0.000000e+00 0.000000e+00  
## [51] 0.000000e+00 0.000000e+00 1.727116e-04 0.000000e+00 0.000000e+00  
## [56] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [61] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [66] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [71] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 8.635579e-05  
## [76] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 8.635579e-05  
## [81] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [86] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [91] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [96] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [101] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [106] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [111] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [116] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [121] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [126] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [131] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [136] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [141] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [146] 0.000000e+00 0.000000e+00 8.635579e-05 0.000000e+00 0.000000e+00  
## [151] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [156] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [161] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [166] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [171] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [176] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [181] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [186] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [191] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [196] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [201] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [206] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [211] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [216] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [221] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [226] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [231] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [236] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [241] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
## [246] 0.000000e+00 8.635579e-05  
##   
## $mids  
## [1] 10 30 50 70 90 110 130 150 170 190 210 230 250 270 290  
## [16] 310 330 350 370 390 410 430 450 470 490 510 530 550 570 590  
## [31] 610 630 650 670 690 710 730 750 770 790 810 830 850 870 890  
## [46] 910 930 950 970 990 1010 1030 1050 1070 1090 1110 1130 1150 1170 1190  
## [61] 1210 1230 1250 1270 1290 1310 1330 1350 1370 1390 1410 1430 1450 1470 1490  
## [76] 1510 1530 1550 1570 1590 1610 1630 1650 1670 1690 1710 1730 1750 1770 1790  
## [91] 1810 1830 1850 1870 1890 1910 1930 1950 1970 1990 2010 2030 2050 2070 2090  
## [106] 2110 2130 2150 2170 2190 2210 2230 2250 2270 2290 2310 2330 2350 2370 2390  
## [121] 2410 2430 2450 2470 2490 2510 2530 2550 2570 2590 2610 2630 2650 2670 2690  
## [136] 2710 2730 2750 2770 2790 2810 2830 2850 2870 2890 2910 2930 2950 2970 2990  
## [151] 3010 3030 3050 3070 3090 3110 3130 3150 3170 3190 3210 3230 3250 3270 3290  
## [166] 3310 3330 3350 3370 3390 3410 3430 3450 3470 3490 3510 3530 3550 3570 3590  
## [181] 3610 3630 3650 3670 3690 3710 3730 3750 3770 3790 3810 3830 3850 3870 3890  
## [196] 3910 3930 3950 3970 3990 4010 4030 4050 4070 4090 4110 4130 4150 4170 4190  
## [211] 4210 4230 4250 4270 4290 4310 4330 4350 4370 4390 4410 4430 4450 4470 4490  
## [226] 4510 4530 4550 4570 4590 4610 4630 4650 4670 4690 4710 4730 4750 4770 4790  
## [241] 4810 4830 4850 4870 4890 4910 4930  
##   
## $xname  
## [1] "liver$Aspartate\_Aminotransferase"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase), length = length(liver$Aspartate\_Aminotransferase)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase),  
 length = length(liver$Aspartate\_Aminotransferase)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase), length = length(liver$Aspartate\_Aminotransferase)),  
 mix.gam.3[["prob"]][2]\*dGA(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase),  
 length = length(liver$Aspartate\_Aminotransferase)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase), length = length(liver$Aspartate\_Aminotransferase)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase),  
 length = length(liver$Aspartate\_Aminotransferase)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam.3[["prob"]][2]\*dRG(seq(min(liver$Aspartate\_Aminotransferase), max(liver$Aspartate\_Aminotransferase),  
 length = length(liver$Aspartate\_Aminotransferase)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")



mix.gm.tb <- data.frame(row.names=c('Gamma Mixture, K=3','Gamma Mixture, K=2', "Generalized Gamma"),  
 AIC=c(mix.gam.3$aic, mix.gam$aic, AIC(aam.GG)),  
 BIC=c(mix.gam.3$sbc, mix.gam$sbc, aam.GG$sbc))  
mix.gm.tb

## AIC BIC  
## Gamma Mixture, K=3 6102.752 6137.642  
## Gamma Mixture, K=2 6178.294 6200.101  
## Generalized Gamma 6072.943 6086.027

We can observe that the AIC value of the mixture of Gamma with k=3 and k=2 has increased, since values are higher than that of the single Gamma distribution. The previous AIC value is 6072.943, whereas the current value which is higher is 6102.752 and the previous BIC value was 6086.027, whereas the current value which is higher is 6137.642. Here we can clearly see that our data fits better in the single Gamma Distribution.

## Total Protiens

Lets analyze the features of Total Protiens variable as follows:

head(liver$Total\_Protiens)

## [1] 6.8 7.5 7.0 6.8 7.3 7.6

length(liver$Total\_Protiens)

## [1] 579

table(liver$Total\_Protiens)

##   
## 2.7 2.8 3 3.6 3.7 3.8 3.9 4 4.1 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5 5.1 5.2 5.3   
## 1 1 1 3 1 2 2 2 2 3 4 4 4 2 3 6 11 10 11 10   
## 5.4 5.5 5.6 5.7 5.8 5.9 6 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7 7.1 7.2 7.3   
## 13 17 18 11 14 14 30 18 24 14 18 14 15 15 28 25 32 22 21 18   
## 7.4 7.5 7.6 7.7 7.8 7.9 8 8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.9 9.2 9.5 9.6   
## 12 15 9 3 9 14 20 6 8 3 3 4 3 1 1 2 1 1

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.

unique(liver$Total\_Protiens)

## [1] 6.8 7.5 7.0 7.3 7.6 6.7 7.4 5.9 8.1 5.8 5.5 6.4 4.3 6.0 5.0 7.2 3.9 5.2 4.9  
## [20] 5.6 6.9 6.2 5.1 6.1 6.5 5.7 6.6 6.3 8.0 4.4 5.3 4.6 4.7 5.4 7.1 4.0 3.7 2.7  
## [39] 3.0 3.8 7.8 4.5 4.1 4.8 7.9 8.5 7.7 8.2 2.8 9.5 9.6 8.3 8.6 8.4 8.9 8.7 3.6  
## [58] 9.2

length(unique(liver$Total\_Protiens))

## [1] 58

min(liver$Total\_Protiens)

## [1] 2.7

max(liver$Total\_Protiens)

## [1] 9.6

Here the total observation of Total\_Protiens is 579 and is a continuous variable that range lies between 2.7-9.. Now, we will see the summary of the Total\_Protiens variable, for further analysis as follows:

summary(liver$Total\_Protiens)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.700 5.800 6.600 6.482 7.200 9.600

sd(liver$Total\_Protiens)

## [1] 1.084641

var(liver$Total\_Protiens)

## [1] 1.176446

v <- c(liver$Total\_Protiens)  
mode <- getMode(v)  
print(mode)

## [1] 7

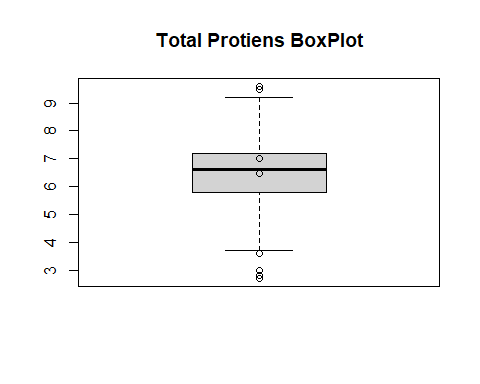
From the above summary we can observe that the Mean (6.482), Median (6.600) and Mode (7) are not equal so the distribution is asymmetrical. Here Mode > Median > Mean so the distribution could be negative and skewed to the left. Now we also confirm this as follows:

library(moments)  
skewness(liver$Total\_Protiens)

## [1] -0.2916746

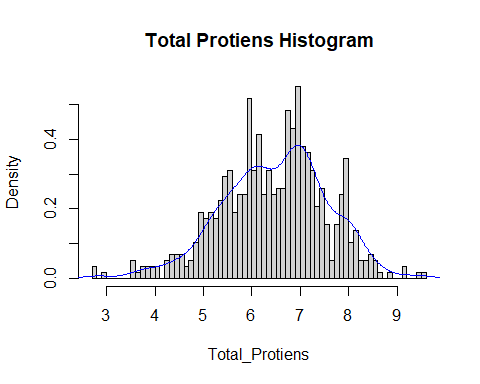
Hence, the skewness is the negative so the distribution is negatively skewed was rightly observed. Also I will plot the Boxplot to comparing the distribution of data across data set as follows:

boxplot(liver$Total\_Protiens, main = "Total Protiens BoxPlot")  
points(mean(liver$Total\_Protiens))  
points(mode)



Box plots are graphical displays of the five number summary, So they describe the Minimum (lowest mean relative) lower whisker, Quartile 1 (It separates the lowest 25% observation to the rest of the observations), Median (Which divides the lower 50% observation from the upper 50% observations), Quartile 3 (It divides the upper 25% observation to the rest of the observations) and Maximum. In the above Box plot diagram, there are some outliers and the upper whisker shows the highest mean relative. Now, I will plot a histogram a graphical display of data using bars of different heights to measures distribution of continuous data as follows:

hist(liver$Total\_Protiens, prob = TRUE, breaks = 58, xlab = "Total\_Protiens", main = "Total Protiens Histogram")  
lines(density(liver$Total\_Protiens), col='blue')



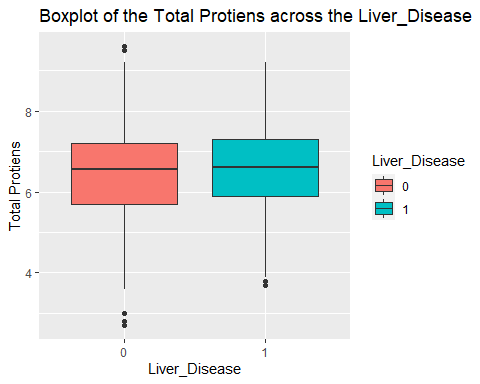
From the above histogram, we can observe that the diagram has many peaks.Using the density lines, I have approximated the histogram graph. Density provides information about the type of distribution under consideration and allows us to determine whether the attribute is distributed normally or not. From the graph above we can see how the Total\_Protiens variable does not seem to fit to a Normal distribution, there is a peak of the frequency distributions around many values and also a left skewed. We have already observed a negative value of -0.2916746, so we will affirm that the distribution is left skewed. To check whether the distribution is Platykurtic or not, we will check this by following R method as follows:

kurtosis(liver$Total\_Protiens)

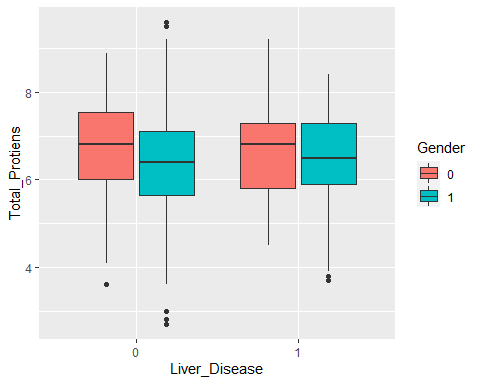
## [1] 3.227065

The distribution is also Leptokurtic, since the value is greater than 3.

ggplot(liver, aes(x = Liver\_Disease, y = Total\_Protiens, fill = Liver\_Disease)) +  
 geom\_boxplot() +  
 ylab("Total Protiens") +  
 ggtitle("Boxplot of the Total Protiens across the Liver\_Disease")



ggplot(liver, aes(Liver\_Disease, Total\_Protiens)) +   
 geom\_boxplot(aes(fill = Gender))

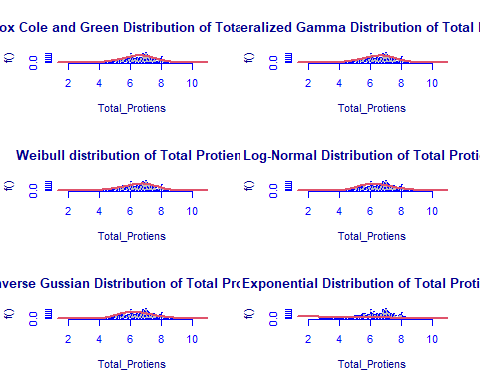


A variable that looks like it has a bell shaped curve. The range for the variable is from 2.7 to 9.6, with a mean of 6.481 and median of 6.6. There does not seem to be much of a difference between the means for the Total Proteins for each response. There is a difference between the responses when you look across the genders. The females have a higher mean for total proteins across the Liver\_Disease group.

### Total Protien Fit for the Data

Now we will try to fit different models to Total Protiens distribution evaluating the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC), The lower are the indexes, the better is the fitting. I will evaluate both the single parameter distributions and mixture distributions as follows:

library(MASS)  
library(gamlss)  
  
  
par(mfrow=c(3,2))  
  
tp.BCCG <- histDist(liver$Total\_Protiens, family=BCCG, nbins = 58, xlab = "Total\_Protiens", main="Box-Cox Cole and Green Distribution of Total Protiens")  
  
tp.GG <- histDist(liver$Total\_Protiens, family=GG, nbins = 58, xlab = "Total\_Protiens", main="Generalized Gamma Distribution of Total Protiens")  
  
tp.WEI <- histDist(liver$Total\_Protiens, family=WEI, nbins = 58, xlab = "Total\_Protiens", main="Weibull distribution of Total Protiens")  
  
tp.LOGNO <- histDist(liver$Total\_Protiens, family=LOGNO, nbins = 58, xlab = "Total\_Protiens", main="Log-Normal Distribution of Total Protiens")  
  
tp.IG <- histDist(liver$Total\_Protiens, family=IG, nbins = 58, xlab = "Total\_Protiens", main = "Inverse Gussian Distribution of Total Protiens")  
  
tp.EXP<- histDist(liver$Total\_Protiens, family=EXP, nbins = 58, xlab = "Total\_Protiens", main = "Exponential Distribution of Total Protiens")



library(Matrix)  
library(glmnet)  
df <- data.frame(Rownames = c("Box-Cox Cole and Green","Generalized Gamma", "Weibull",  
 "Log-Normal", "Inverse Gussian", "Exponential"),  
 AIC = c(AIC(tp.BCCG), AIC(tp.GG), AIC(tp.WEI), AIC(tp.LOGNO),  
 AIC(tp.IG), AIC(tp.EXP)),  
 BIC = c(tp.BCCG$sbc, tp.GG$sbc, tp.WEI$sbc, tp.LOGNO$sbc,  
 tp.IG$sbc, tp.EXP$sbc),  
 df = c(tp.BCCG$df.fit, tp.GG$df.fit, tp.WEI$df.fit,  
 tp.LOGNO$df.fit, tp.IG$df.fit, tp.EXP$df.fit),  
 LogLike = c(logLik(tp.BCCG), logLik(tp.GG), logLik(tp.WEI),  
 logLik(tp.LOGNO), logLik(tp.IG), logLik(tp.EXP)))  
df

## Rownames AIC BIC df LogLike  
## 1 Box-Cox Cole and Green 1734.355 1747.439 3 -864.1775  
## 2 Generalized Gamma 1735.067 1748.151 3 -864.5336  
## 3 Weibull 1738.050 1746.773 2 -867.0252  
## 4 Log-Normal 1804.629 1813.352 2 -900.3145  
## 5 Inverse Gussian 1808.334 1817.057 2 -902.1670  
## 6 Exponential 3324.281 3328.642 1 -1661.1404

As we can see, the model with the highest log likelihood (-864.1775) and the lowest AIC (1734.355) and BIC (1747.439) is the Box-Cox Cole and Green Distribution. Therefore, based on the greatest likelihood technique, our data fits better in the Box-Cox Cole and Green Distribution. Now, I will also compare two models by means of the Likelihood ratio test as we performed above as follows:

library(zoo)  
library(lmtest)  
lrtest(tp.BCCG, tp.EXP)

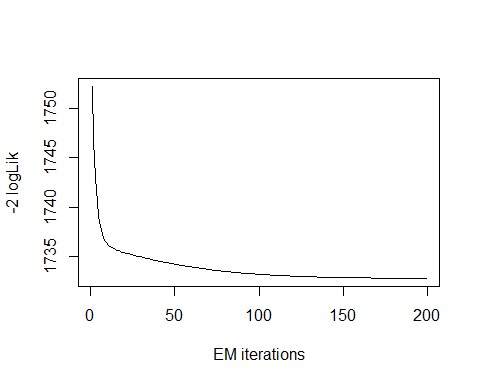
## Likelihood ratio test  
##   
## Model 1: gamlssML(formula = liver$Total\_Protiens, family = "BCCG")  
## Model 2: gamlssML(formula = liver$Total\_Protiens, family = "EXP")  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -864.18   
## 2 1 -1661.14 -2 1593.9 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Under the null hypothesis, we compare the Box-Cox Cole and Green distribution to the Exponential distribution under the alternative hypothesis. At any level of significance, we can reject the null hypothesis. Because our distributions fit better in the Box-Cox Cole and Green model, we’ll use it.

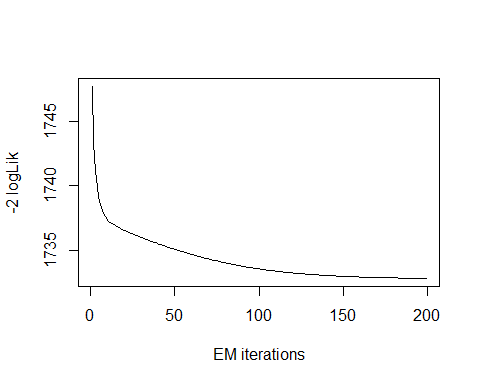
### Distributions Mixture

Now, I will also apply the fitting criteria for the distributions with Gamma mixture as follows:

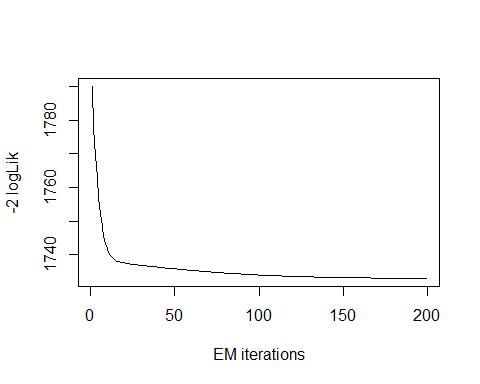
library(gamlss.mx)  
mix.gam <- gamlssMXfits(n = 5, liver$Total\_Protiens~1, family = GA, K = 2, data = liver)



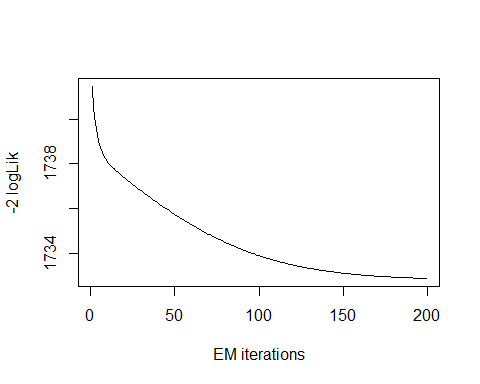
## model= 1



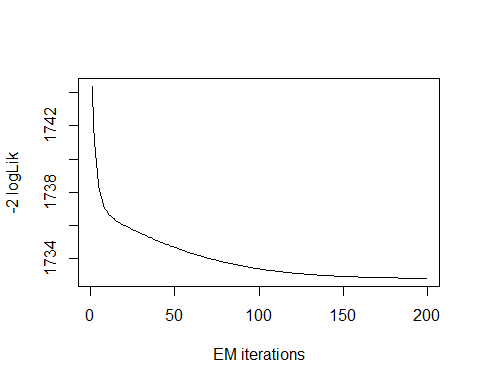
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam)

##   
## Mixing Family: c("GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Total\_Protiens ~ 1, family = GA,   
## K = 2, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 1.901   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -1.978   
## Mu Coefficients for model: 2   
## (Intercept)   
## 1.682   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -1.493   
##   
## Estimated probabilities: 0.8390968 0.1609032   
##   
## Degrees of Freedom for the fit: 5 Residual Deg. of Freedom 574   
## Global Deviance: 1732.82   
## AIC: 1742.82   
## SBC: 1764.63

We can observe that the AIC value of the mixture of Gamma has increased, since it is higher than that of the single Gamma distribution. The current AIC value is 1742.81, whereas the previous value was 1734.355 and the current BIC value is 1764.62, whereas the previous value was 1747.439.

logLik(mix.gam)

## 'log Lik.' -866.4106 (df=5)

mix.gam$prob

## [1] 0.8390968 0.1609032

fitted(mix.gam, "mu")[1]

## [1] 6.482017

fitted(mix.gam, "sigma")[2]

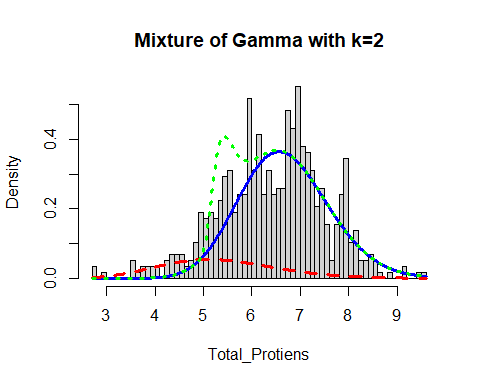
## [1] 6.482017

hist(liver$Total\_Protiens, breaks = 58, xlab = "Total\_Protiens", main="Mixture of Gamma with k=2", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Total\_Protiens, breaks = 58, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Total\_Protiens, breaks = 58, freq = FALSE, :  
## argument 'freq' is not made use of

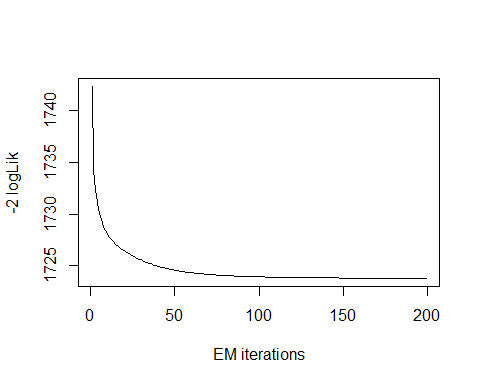
## $breaks  
## [1] 2.7 2.8 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3 4.4 4.5  
## [20] 4.6 4.7 4.8 4.9 5.0 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 6.0 6.1 6.2 6.3 6.4  
## [39] 6.5 6.6 6.7 6.8 6.9 7.0 7.1 7.2 7.3 7.4 7.5 7.6 7.7 7.8 7.9 8.0 8.1 8.2 8.3  
## [58] 8.4 8.5 8.6 8.7 8.8 8.9 9.0 9.1 9.2 9.3 9.4 9.5 9.6  
##   
## $counts  
## [1] 2 0 1 0 0 0 0 0 3 1 2 2 2 2 0 3 4 4 4 2 3 6 11 10 11  
## [26] 10 13 17 18 11 14 14 30 18 24 14 18 14 15 15 28 25 32 22 21 18 12 15 9 3  
## [51] 9 14 20 6 8 3 3 4 3 1 0 1 0 0 2 0 0 1 1  
##   
## $density  
## [1] 0.03454231 0.00000000 0.01727116 0.00000000 0.00000000 0.00000000  
## [7] 0.00000000 0.00000000 0.05181347 0.01727116 0.03454231 0.03454231  
## [13] 0.03454231 0.03454231 0.00000000 0.05181347 0.06908463 0.06908463  
## [19] 0.06908463 0.03454231 0.05181347 0.10362694 0.18998273 0.17271157  
## [25] 0.18998273 0.17271157 0.22452504 0.29360967 0.31088083 0.18998273  
## [31] 0.24179620 0.24179620 0.51813472 0.31088083 0.41450777 0.24179620  
## [37] 0.31088083 0.24179620 0.25906736 0.25906736 0.48359240 0.43177893  
## [43] 0.55267703 0.37996546 0.36269430 0.31088083 0.20725389 0.25906736  
## [49] 0.15544041 0.05181347 0.15544041 0.24179620 0.34542314 0.10362694  
## [55] 0.13816926 0.05181347 0.05181347 0.06908463 0.05181347 0.01727116  
## [61] 0.00000000 0.01727116 0.00000000 0.00000000 0.03454231 0.00000000  
## [67] 0.00000000 0.01727116 0.01727116  
##   
## $mids  
## [1] 2.75 2.85 2.95 3.05 3.15 3.25 3.35 3.45 3.55 3.65 3.75 3.85 3.95 4.05 4.15  
## [16] 4.25 4.35 4.45 4.55 4.65 4.75 4.85 4.95 5.05 5.15 5.25 5.35 5.45 5.55 5.65  
## [31] 5.75 5.85 5.95 6.05 6.15 6.25 6.35 6.45 6.55 6.65 6.75 6.85 6.95 7.05 7.15  
## [46] 7.25 7.35 7.45 7.55 7.65 7.75 7.85 7.95 8.05 8.15 8.25 8.35 8.45 8.55 8.65  
## [61] 8.75 8.85 8.95 9.05 9.15 9.25 9.35 9.45 9.55  
##   
## $xname  
## [1] "liver$Total\_Protiens"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens), length = length(liver$Total\_Protiens)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens),  
 length = length(liver$Total\_Protiens)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens), length = length(liver$Total\_Protiens)),  
 mix.gam[["prob"]][2]\*dGA(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens),  
 length = length(liver$Total\_Protiens)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens), length = length(liver$Total\_Protiens)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens),  
 length = length(liver$Total\_Protiens)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam[["prob"]][2]\*dRG(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens),  
 length = length(liver$Total\_Protiens)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")

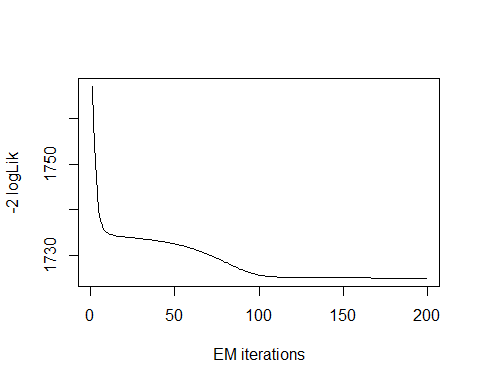


Now, I will again apply the fitting criteria for the distributions with Gamma mixture with the k = 3 as follows:

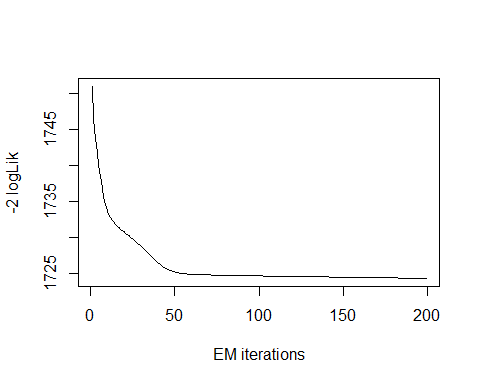
mix.gam.3 <- gamlssMXfits(n = 5, liver$Total\_Protiens~1, family = GA, K = 3, data = liver)



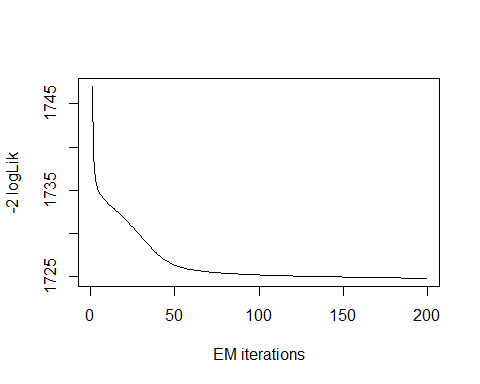
## model= 1



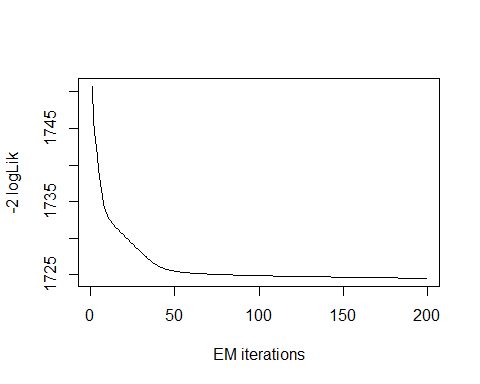
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam.3)

##   
## Mixing Family: c("GA", "GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Total\_Protiens ~ 1, family = GA,   
## K = 3, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 1.735   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -2.44   
## Mu Coefficients for model: 2   
## (Intercept)   
## 1.953   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -2.254   
## Mu Coefficients for model: 3   
## (Intercept)   
## 1.689   
## Sigma Coefficients for model: 3   
## (Intercept)   
## -1.436   
##   
## Estimated probabilities: 0.2289222 0.616795 0.1542828   
##   
## Degrees of Freedom for the fit: 8 Residual Deg. of Freedom 571   
## Global Deviance: 1723.79   
## AIC: 1739.79   
## SBC: 1774.68

logLik(mix.gam.3)

## 'log Lik.' -861.8965 (df=8)

mix.gam.3$prob

## [1] 0.2289222 0.6167950 0.1542828

fitted(mix.gam.3, "mu")[1]

## [1] 6.481988

fitted(mix.gam.3, "sigma")[2]

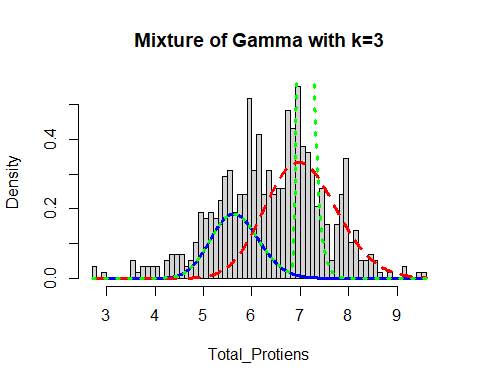
## [1] 6.481988

hist(liver$Total\_Protiens, breaks = 58, xlab = "Total\_Protiens", main="Mixture of Gamma with k=3", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam.3[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam.3[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam.3[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam.3[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Total\_Protiens, breaks = 58, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Total\_Protiens, breaks = 58, freq = FALSE, :  
## argument 'freq' is not made use of

## $breaks  
## [1] 2.7 2.8 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3 4.4 4.5  
## [20] 4.6 4.7 4.8 4.9 5.0 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 6.0 6.1 6.2 6.3 6.4  
## [39] 6.5 6.6 6.7 6.8 6.9 7.0 7.1 7.2 7.3 7.4 7.5 7.6 7.7 7.8 7.9 8.0 8.1 8.2 8.3  
## [58] 8.4 8.5 8.6 8.7 8.8 8.9 9.0 9.1 9.2 9.3 9.4 9.5 9.6  
##   
## $counts  
## [1] 2 0 1 0 0 0 0 0 3 1 2 2 2 2 0 3 4 4 4 2 3 6 11 10 11  
## [26] 10 13 17 18 11 14 14 30 18 24 14 18 14 15 15 28 25 32 22 21 18 12 15 9 3  
## [51] 9 14 20 6 8 3 3 4 3 1 0 1 0 0 2 0 0 1 1  
##   
## $density  
## [1] 0.03454231 0.00000000 0.01727116 0.00000000 0.00000000 0.00000000  
## [7] 0.00000000 0.00000000 0.05181347 0.01727116 0.03454231 0.03454231  
## [13] 0.03454231 0.03454231 0.00000000 0.05181347 0.06908463 0.06908463  
## [19] 0.06908463 0.03454231 0.05181347 0.10362694 0.18998273 0.17271157  
## [25] 0.18998273 0.17271157 0.22452504 0.29360967 0.31088083 0.18998273  
## [31] 0.24179620 0.24179620 0.51813472 0.31088083 0.41450777 0.24179620  
## [37] 0.31088083 0.24179620 0.25906736 0.25906736 0.48359240 0.43177893  
## [43] 0.55267703 0.37996546 0.36269430 0.31088083 0.20725389 0.25906736  
## [49] 0.15544041 0.05181347 0.15544041 0.24179620 0.34542314 0.10362694  
## [55] 0.13816926 0.05181347 0.05181347 0.06908463 0.05181347 0.01727116  
## [61] 0.00000000 0.01727116 0.00000000 0.00000000 0.03454231 0.00000000  
## [67] 0.00000000 0.01727116 0.01727116  
##   
## $mids  
## [1] 2.75 2.85 2.95 3.05 3.15 3.25 3.35 3.45 3.55 3.65 3.75 3.85 3.95 4.05 4.15  
## [16] 4.25 4.35 4.45 4.55 4.65 4.75 4.85 4.95 5.05 5.15 5.25 5.35 5.45 5.55 5.65  
## [31] 5.75 5.85 5.95 6.05 6.15 6.25 6.35 6.45 6.55 6.65 6.75 6.85 6.95 7.05 7.15  
## [46] 7.25 7.35 7.45 7.55 7.65 7.75 7.85 7.95 8.05 8.15 8.25 8.35 8.45 8.55 8.65  
## [61] 8.75 8.85 8.95 9.05 9.15 9.25 9.35 9.45 9.55  
##   
## $xname  
## [1] "liver$Total\_Protiens"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens), length = length(liver$Total\_Protiens)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens),  
 length = length(liver$Total\_Protiens)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens), length = length(liver$Total\_Protiens)),  
 mix.gam.3[["prob"]][2]\*dGA(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens),  
 length = length(liver$Total\_Protiens)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens), length = length(liver$Total\_Protiens)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens),  
 length = length(liver$Total\_Protiens)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam.3[["prob"]][2]\*dRG(seq(min(liver$Total\_Protiens), max(liver$Total\_Protiens),  
 length = length(liver$Total\_Protiens)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")



mix.gm.tb <- data.frame(row.names=c('Gamma Mixture, K=3','Gamma Mixture, K=2', "Generalized Gamma"),  
 AIC=c(mix.gam.3$aic, mix.gam$aic, AIC(tp.GG)),  
 BIC=c(mix.gam.3$sbc, mix.gam$sbc, tp.GG$sbc))  
mix.gm.tb

## AIC BIC  
## Gamma Mixture, K=3 1739.793 1774.683  
## Gamma Mixture, K=2 1742.821 1764.628  
## Generalized Gamma 1735.067 1748.151

We can observe that the AIC value of the mixture of Gamma with k=3 and k=2 has increased, since values are higher than that of the single Gamma distribution. The previous AIC value is 1735.067, whereas the current value which is higher is 1739.865 and the previous BIC value was 1748.151, whereas the current value which is higher is 1774.755. Here we can clearly see that our data fits better in the single Gamma Distribution.

## Albumin

Lets explore the Albumin variable:

head(liver$Albumin)

## [1] 3.3 3.2 3.3 3.4 2.4 4.4

length(liver$Albumin)

## [1] 579

table(liver$Albumin)

##   
## 0.9 1 1.4 1.5 1.6 1.7 1.8 1.9 2 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3 3.1   
## 2 1 3 3 8 3 12 7 21 14 12 12 17 24 21 23 18 29 45 27   
## 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5 5.5   
## 26 21 21 23 18 21 15 24 37 16 12 14 8 6 4 3 1 4 1 2

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.

unique(liver$Albumin)

## [1] 3.3 3.2 3.4 2.4 4.4 3.5 3.6 4.1 2.7 3.0 2.3 3.1 2.6 1.6 3.9 4.0 1.9 1.5 2.9  
## [20] 2.0 2.2 2.8 1.8 2.5 2.1 3.7 3.8 4.3 1.7 4.2 4.5 0.9 1.4 4.7 5.5 4.9 4.6 5.0  
## [39] 4.8 1.0

length(unique(liver$Albumin))

## [1] 40

min(liver$Albumin)

## [1] 0.9

max(liver$Albumin)

## [1] 5.5

Here the total observation of Albumin is 579 and is a continuous variable that range lies between 0.9-5.5. Now, we will see the summary of the Albumin variable, for further analysis as follows:

summary(liver$Albumin)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.900 2.600 3.100 3.139 3.800 5.500

sd(liver$Albumin)

## [1] 0.7944347

var(liver$Albumin)

## [1] 0.6311265

v <- c(liver$Albumin)  
mode <- getMode(v)  
print(mode)

## [1] 3

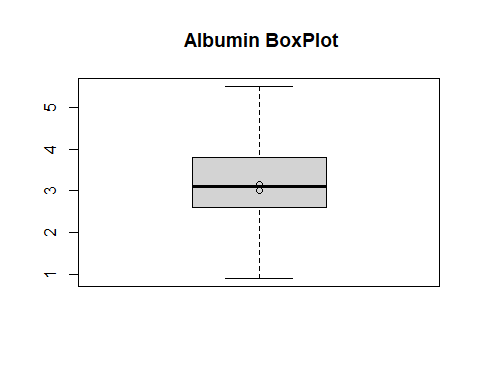
From the above summary we can observe that the Mean (3.139), Median (3.100) and Mode (3) are not equal so the distribution is asymmetrical. Here Mean > Median > Mode so the distribution could be positive and skewed to the right. Now we also confirm this as follows:

library(moments)  
skewness(liver$Albumin)

## [1] -0.04839028

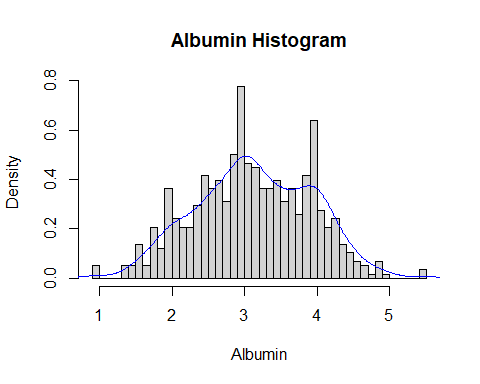
Hence, the skewness is the negative so the distribution is negatively skewed. Also I will plot the Boxplot to comparing the distribution of data across dataset as follows:

boxplot(liver$Albumin, main = " Albumin BoxPlot")  
points(mean(liver$Albumin))  
points(mode)



Box plots are graphical displays of the five number summary, So they describe the Minimum (lowest mean relative) lower whisker, Quartile 1 (It separates the lowest 25% observation to the rest of the observations), Median (Which divides the lower 50% observation from the upper 50% observations), Quartile 3 (It divides the upper 25% observation to the rest of the observations) and Maximum. In the above Box plot diagram, there are no outliers and the upper whisker shows the maximum. Now, I will plot a histogram a graphical display of data using bars of different heights to measures distribution of continuous data as follows:

hist(liver$Albumin, prob = TRUE, breaks = 40, xlab = "Albumin", main = "Albumin Histogram")  
lines(density(liver$Albumin), col='blue')



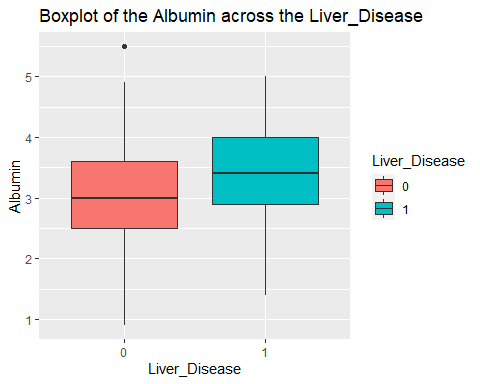
From the above histogram, we can observe that the diagram has many peaks.Using the density lines, I have approximated the histogram graph. Density provides information about the type of distribution under consideration and allows us to determine whether the attribute is distributed normally or not. From the graph above we can see how the Albumin variable does not seem to fit to a Normal distribution, there is a peak of the frequency distributions around many values and also a left skewed. We have already observed a negative value of -0.04839028, so we will affirm that the distribution is left skewed. To check whether the distribution is Platykurtic or not, we will check this by following R method as follows:

kurtosis(liver$Albumin)

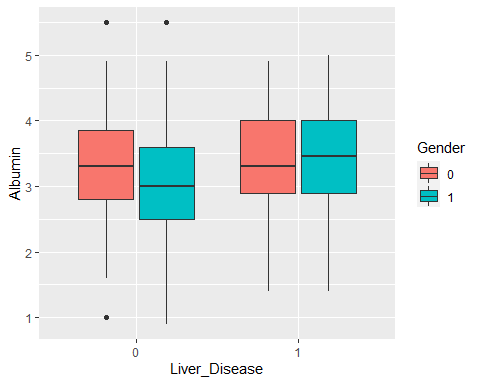
## [1] 2.602133

The distribution is also Platykurtic, since the value is less than 3.

ggplot(liver, aes(x = Liver\_Disease, y = Albumin, fill = Liver\_Disease)) +  
 geom\_boxplot() +  
 ylab("Albumin") +  
 ggtitle("Boxplot of the Albumin across the Liver\_Disease")



ggplot(liver, aes(Liver\_Disease, Albumin)) +   
 geom\_boxplot(aes(fill = Gender))

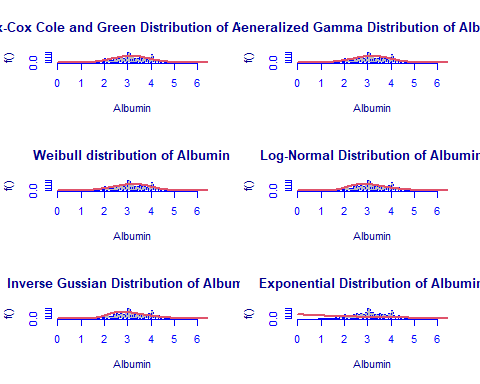


Range from 0.9 to 5.5, mean of 3.139, and median of 3.100. Bell shaped density for the Albumin values. The mean is higher for Liver\_Disease = 1 and the range is larger when the Liver\_Disease = 0. There is a clear difference in the means for the Albumin values across the genders and Liver\_Disease. The mean is higher for females when the Liver\_Disease = 0 and is higher for males when the Liver\_Disease = 1.

### Albumin Fit for the Data

Now we will try to fit different models to Albumin distribution evaluating the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC), The lower are the indexes, the better is the fitting. I will evaluate both the single parameter distributions and mixture distributions as follows:

library(MASS)  
library(gamlss)  
  
  
par(mfrow=c(3,2))  
  
a.BCCG <- histDist(liver$Albumin, family=BCCG, nbins = 40, xlab = "Albumin", main="Box-Cox Cole and Green Distribution of Albumin")  
  
a.GG <- histDist(liver$Albumin, family=GG, nbins = 40, xlab = "Albumin", main="Generalized Gamma Distribution of Albumin")  
  
a.WEI <- histDist(liver$Albumin, family=WEI, nbins = 40, xlab = "Albumin", main="Weibull distribution of Albumin")  
  
a.LOGNO <- histDist(liver$Albumin, family=LOGNO, nbins = 40, xlab = "Albumin", main="Log-Normal Distribution of Albumin")  
  
a.IG <- histDist(liver$Albumin, family=IG, nbins = 40, xlab = "Albumin", main = "Inverse Gussian Distribution of Albumin")  
  
a.EXP<- histDist(liver$Albumin, family=EXP, nbins = 40, xlab = "Albumin", main = "Exponential Distribution of Albumin")



library(Matrix)  
library(glmnet)  
df <- data.frame(Rownames = c("Box-Cox Cole and Green","Generalized Gamma", "Weibull",  
 "Log-Normal", "Inverse Gussian", "Exponential"),  
 AIC = c(AIC(a.BCCG), AIC(a.GG), AIC(a.WEI), AIC(a.LOGNO),  
 AIC(a.IG), AIC(a.EXP)),  
 BIC = c(a.BCCG$sbc, a.GG$sbc, a.WEI$sbc, a.LOGNO$sbc,  
 a.IG$sbc, a.EXP$sbc),  
 df = c(a.BCCG$df.fit, a.GG$df.fit, a.WEI$df.fit,  
 a.LOGNO$df.fit, a.IG$df.fit, a.EXP$df.fit),  
 LogLike = c(logLik(a.BCCG), logLik(a.GG), logLik(a.WEI),  
 logLik(a.LOGNO), logLik(a.IG), logLik(a.EXP)))  
df

## Rownames AIC BIC df LogLike  
## 1 Box-Cox Cole and Green 1381.580 1394.663 3 -687.7898  
## 2 Generalized Gamma 1377.845 1390.929 3 -685.9226  
## 3 Weibull 1377.253 1385.976 2 -686.6266  
## 4 Log-Normal 1442.531 1451.253 2 -719.2653  
## 5 Inverse Gussian 1448.224 1456.947 2 -722.1121  
## 6 Exponential 2484.462 2488.823 1 -1241.2311

As we can see, the model with the highest log likelihood (-685.9226) and the lowest AIC (1377.845) and BIC (1390.929) is the Generalized Gamma Distribution. Therefore, based on the greatest likelihood technique, our data fits better in the Generalized Gamma Distribution. Now, I will also compare two models by means of the Likelihood ratio test as we performed above as follows:

library(zoo)  
library(lmtest)  
lrtest(a.GG, a.EXP)

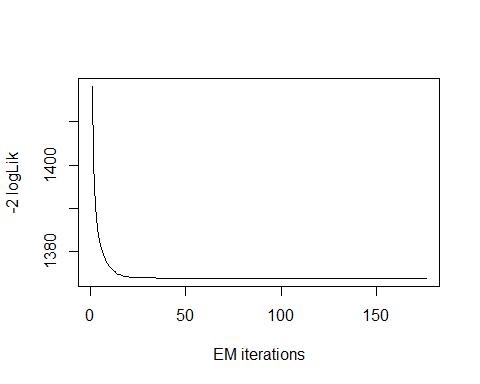
## Likelihood ratio test  
##   
## Model 1: gamlssML(formula = liver$Albumin, family = "GG")  
## Model 2: gamlssML(formula = liver$Albumin, family = "EXP")  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -685.92   
## 2 1 -1241.23 -2 1110.6 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Under the null hypothesis, we compare the Generalized Gamma distribution to the Exponential distribution under the alternative hypothesis. At any level of significance, we can reject the null hypothesis. Because our distributions fit better in the Generalized Gamma model, we’ll use it.

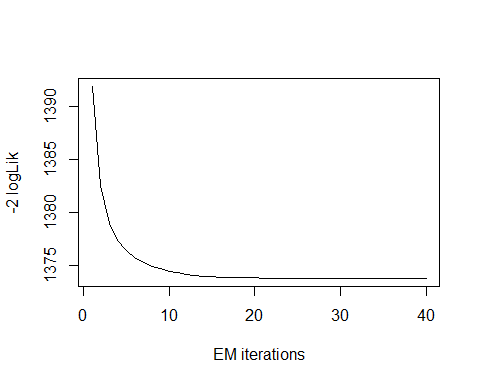
### Distributions Mixture

Now, I will also apply the fitting criteria for the distributions with Gamma mixture as follows:

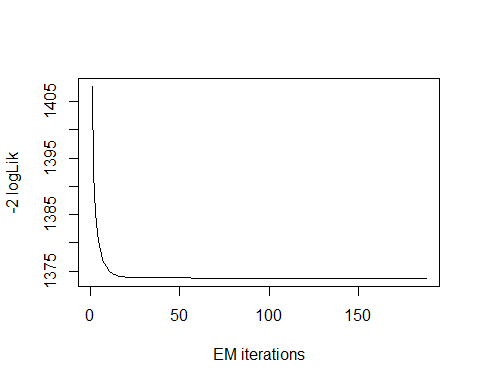
library(gamlss.mx)  
mix.gam <- gamlssMXfits(n = 5, liver$Albumin~1, family = GA, K = 2, data = liver)



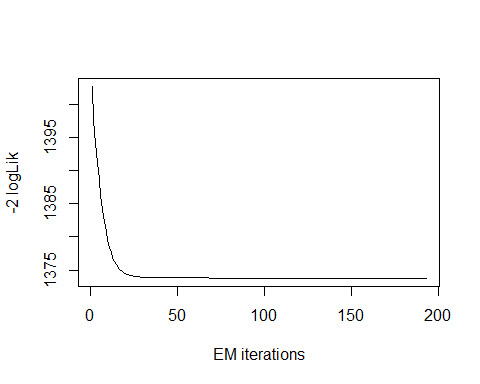
## model= 1



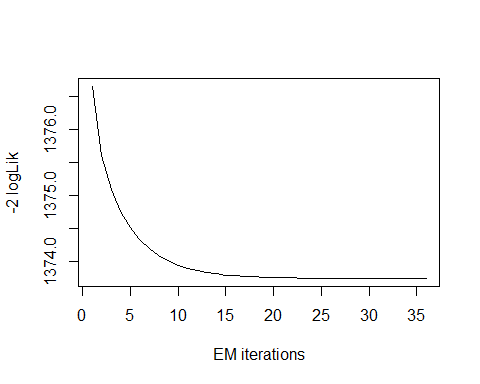
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam)

##   
## Mixing Family: c("GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Albumin ~ 1, family = GA,   
## K = 2, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 1.007   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -1.305   
## Mu Coefficients for model: 2   
## (Intercept)   
## 1.286   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -1.848   
##   
## Estimated probabilities: 0.5431728 0.4568272   
##   
## Degrees of Freedom for the fit: 5 Residual Deg. of Freedom 574   
## Global Deviance: 1373.75   
## AIC: 1383.75   
## SBC: 1405.55

We can observe that the AIC value of the mixture of Gamma has improved, since it is higher than that of the single Gamma distribution. The current AIC value is 1383.75, whereas the previous value was 1377.845and the current BIC value is 1405.55, whereas the previous value was 1390.929.

logLik(mix.gam)

## 'log Lik.' -686.8726 (df=5)

mix.gam$prob

## [1] 0.5431728 0.4568272

fitted(mix.gam, "mu")[1]

## [1] 3.138593

fitted(mix.gam, "sigma")[2]

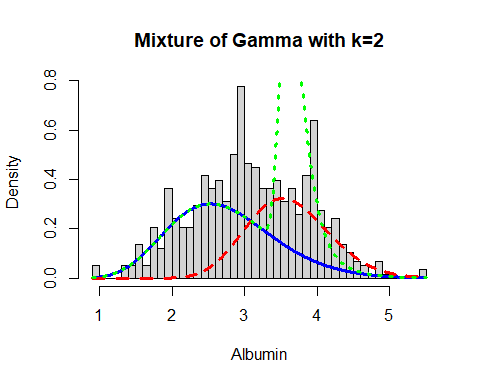
## [1] 3.138593

hist(liver$Albumin, breaks = 40, xlab = "Albumin", main="Mixture of Gamma with k=2", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Albumin, breaks = 40, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Albumin, breaks = 40, freq = FALSE, plot = FALSE):  
## argument 'freq' is not made use of

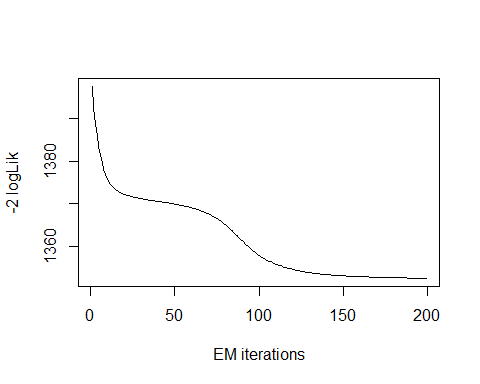
## $breaks  
## [1] 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7  
## [20] 2.8 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3 4.4 4.5 4.6  
## [39] 4.7 4.8 4.9 5.0 5.1 5.2 5.3 5.4 5.5  
##   
## $counts  
## [1] 3 0 0 0 3 3 8 3 12 7 21 14 12 12 17 24 21 23 18 29 45 27 26 21 21  
## [26] 23 18 21 15 24 37 16 12 14 8 6 4 3 1 4 1 0 0 0 0 2  
##   
## $density  
## [1] 0.05181347 0.00000000 0.00000000 0.00000000 0.05181347 0.05181347  
## [7] 0.13816926 0.05181347 0.20725389 0.12089810 0.36269430 0.24179620  
## [13] 0.20725389 0.20725389 0.29360967 0.41450777 0.36269430 0.39723661  
## [19] 0.31088083 0.50086356 0.77720207 0.46632124 0.44905009 0.36269430  
## [25] 0.36269430 0.39723661 0.31088083 0.36269430 0.25906736 0.41450777  
## [31] 0.63903282 0.27633851 0.20725389 0.24179620 0.13816926 0.10362694  
## [37] 0.06908463 0.05181347 0.01727116 0.06908463 0.01727116 0.00000000  
## [43] 0.00000000 0.00000000 0.00000000 0.03454231  
##   
## $mids  
## [1] 0.95 1.05 1.15 1.25 1.35 1.45 1.55 1.65 1.75 1.85 1.95 2.05 2.15 2.25 2.35  
## [16] 2.45 2.55 2.65 2.75 2.85 2.95 3.05 3.15 3.25 3.35 3.45 3.55 3.65 3.75 3.85  
## [31] 3.95 4.05 4.15 4.25 4.35 4.45 4.55 4.65 4.75 4.85 4.95 5.05 5.15 5.25 5.35  
## [46] 5.45  
##   
## $xname  
## [1] "liver$Albumin"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Albumin), max(liver$Albumin), length = length(liver$Albumin)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Albumin), max(liver$Albumin),  
 length = length(liver$Albumin)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Albumin), max(liver$Albumin), length = length(liver$Albumin)),  
 mix.gam[["prob"]][2]\*dGA(seq(min(liver$Albumin), max(liver$Albumin),  
 length = length(liver$Albumin)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Albumin), max(liver$Albumin), length = length(liver$Albumin)),  
 mix.gam[["prob"]][1]\*dGA(seq(min(liver$Albumin), max(liver$Albumin),  
 length = length(liver$Albumin)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam[["prob"]][2]\*dRG(seq(min(liver$Albumin), max(liver$Albumin),  
 length = length(liver$Albumin)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")

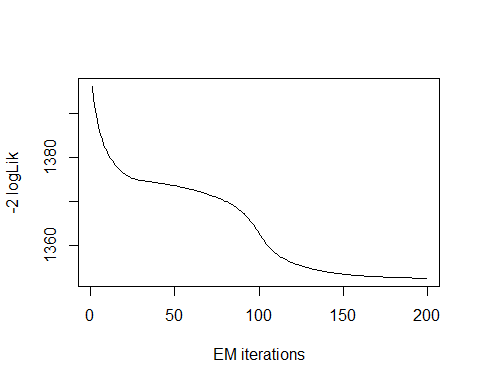


Now, I will again apply the fitting criteria for the distributions with Gamma mixture with the k = 3 as follows:

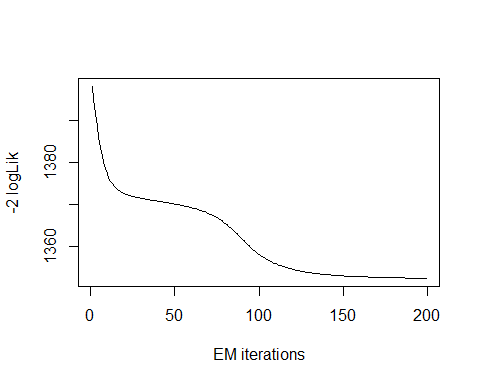
mix.gam.3 <- gamlssMXfits(n = 5, liver$Albumin~1, family = GA, K = 3, data = liver)



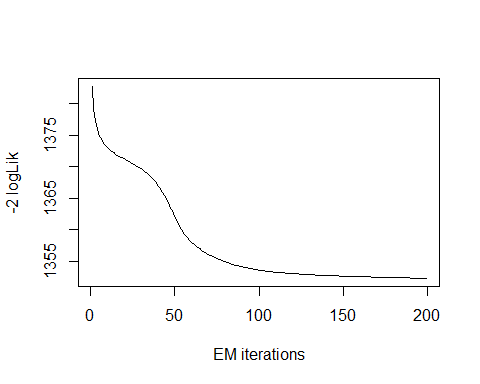
## model= 1



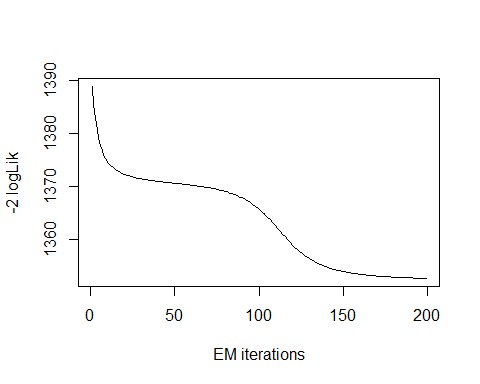
## model= 2



## model= 3



## model= 4



## model= 5

print(mix.gam.3)

##   
## Mixing Family: c("GA", "GA", "GA")   
##   
## Fitting method: EM algorithm   
##   
## Call: gamlssMX(formula = liver$Albumin ~ 1, family = GA,   
## K = 3, data = liver)   
##   
## Mu Coefficients for model: 1   
## (Intercept)   
## 1.379   
## Sigma Coefficients for model: 1   
## (Intercept)   
## -2.594   
## Mu Coefficients for model: 2   
## (Intercept)   
## 1.123   
## Sigma Coefficients for model: 2   
## (Intercept)   
## -2.659   
## Mu Coefficients for model: 3   
## (Intercept)   
## 1.047   
## Sigma Coefficients for model: 3   
## (Intercept)   
## -1.263   
##   
## Estimated probabilities: 0.2266227 0.1564675 0.6169098   
##   
## Degrees of Freedom for the fit: 8 Residual Deg. of Freedom 571   
## Global Deviance: 1352.27   
## AIC: 1368.27   
## SBC: 1403.17

logLik(mix.gam.3)

## 'log Lik.' -676.1375 (df=8)

mix.gam.3$prob

## [1] 0.2266227 0.1564675 0.6169098

fitted(mix.gam.3, "mu")[1]

## [1] 3.138605

fitted(mix.gam.3, "sigma")[2]

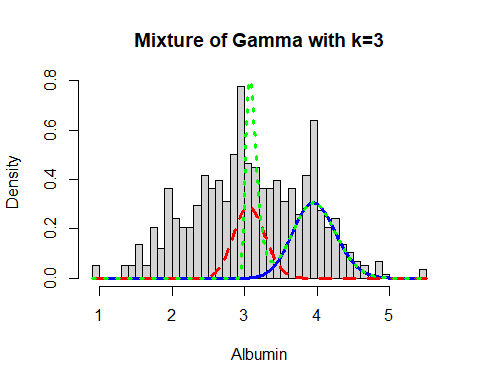
## [1] 3.138605

hist(liver$Albumin, breaks = 40, xlab = "Albumin", main="Mixture of Gamma with k=3", freq = FALSE)  
  
mu.hat1 <- exp(mix.gam.3[["models"]][[1]][["mu.coefficients"]])  
  
sigma.hat1 <- exp(mix.gam.3[["models"]][[1]][["sigma.coefficients"]])  
  
mu.hat2 <- exp(mix.gam.3[["models"]][[2]][["mu.coefficients"]])  
  
sigma.hat2 <- exp(mix.gam.3[["models"]][[2]][["sigma.coefficients"]])  
  
hist(liver$Albumin, breaks = 40, freq = FALSE, plot = FALSE)

## Warning in hist.default(liver$Albumin, breaks = 40, freq = FALSE, plot = FALSE):  
## argument 'freq' is not made use of

## $breaks  
## [1] 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7  
## [20] 2.8 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3 4.4 4.5 4.6  
## [39] 4.7 4.8 4.9 5.0 5.1 5.2 5.3 5.4 5.5  
##   
## $counts  
## [1] 3 0 0 0 3 3 8 3 12 7 21 14 12 12 17 24 21 23 18 29 45 27 26 21 21  
## [26] 23 18 21 15 24 37 16 12 14 8 6 4 3 1 4 1 0 0 0 0 2  
##   
## $density  
## [1] 0.05181347 0.00000000 0.00000000 0.00000000 0.05181347 0.05181347  
## [7] 0.13816926 0.05181347 0.20725389 0.12089810 0.36269430 0.24179620  
## [13] 0.20725389 0.20725389 0.29360967 0.41450777 0.36269430 0.39723661  
## [19] 0.31088083 0.50086356 0.77720207 0.46632124 0.44905009 0.36269430  
## [25] 0.36269430 0.39723661 0.31088083 0.36269430 0.25906736 0.41450777  
## [31] 0.63903282 0.27633851 0.20725389 0.24179620 0.13816926 0.10362694  
## [37] 0.06908463 0.05181347 0.01727116 0.06908463 0.01727116 0.00000000  
## [43] 0.00000000 0.00000000 0.00000000 0.03454231  
##   
## $mids  
## [1] 0.95 1.05 1.15 1.25 1.35 1.45 1.55 1.65 1.75 1.85 1.95 2.05 2.15 2.25 2.35  
## [16] 2.45 2.55 2.65 2.75 2.85 2.95 3.05 3.15 3.25 3.35 3.45 3.55 3.65 3.75 3.85  
## [31] 3.95 4.05 4.15 4.25 4.35 4.45 4.55 4.65 4.75 4.85 4.95 5.05 5.15 5.25 5.35  
## [46] 5.45  
##   
## $xname  
## [1] "liver$Albumin"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

lines(seq(min(liver$Albumin), max(liver$Albumin), length = length(liver$Albumin)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Albumin), max(liver$Albumin),  
 length = length(liver$Albumin)), mu = mu.hat1, sigma = sigma.hat1),  
 lty=1, lwd=3, col="blue")  
  
lines(seq(min(liver$Albumin), max(liver$Albumin), length = length(liver$Albumin)),  
 mix.gam.3[["prob"]][2]\*dGA(seq(min(liver$Albumin), max(liver$Albumin),  
 length = length(liver$Albumin)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 2, lwd = 3, col = "red")  
  
lines(seq(min(liver$Albumin), max(liver$Albumin), length = length(liver$Albumin)),  
 mix.gam.3[["prob"]][1]\*dGA(seq(min(liver$Albumin), max(liver$Albumin),  
 length = length(liver$Albumin)), mu = mu.hat1, sigma = sigma.hat1)+  
 mix.gam.3[["prob"]][2]\*dRG(seq(min(liver$Albumin), max(liver$Albumin),  
 length = length(liver$Albumin)), mu = mu.hat2, sigma = sigma.hat2),  
 lty = 3, lwd = 3, col = "green")



mix.gm.tb <- data.frame(row.names=c('Gamma Mixture, K=3','Gamma Mixture, K=2', "Generalized Gamma"),  
 AIC=c(mix.gam.3$aic, mix.gam$aic, AIC(a.GG)),  
 BIC=c(mix.gam.3$sbc, mix.gam$sbc, a.GG$sbc))  
mix.gm.tb

## AIC BIC  
## Gamma Mixture, K=3 1368.275 1403.165  
## Gamma Mixture, K=2 1383.745 1405.552  
## Generalized Gamma 1377.845 1390.929

We can observe that the AIC value of the mixture of Gamma with k=3 has increased, since values are higher than that of the single Gamma distribution and Gamma Mixture k=2 . The previous AIC value is 1377.845, whereas the current value which is lower is 1368.254. Here we can clearly see that our data fits better in the Gamma Mixture with k=3 Distribution.

## Albumin and Globulin Ratio

Lets explore the Albumin and Globulin Ratio variable:

head(liver$Albumin\_and\_Globulin\_Ratio)

## [1] 0.90 0.74 0.89 1.00 0.40 1.30

length(liver$Albumin\_and\_Globulin\_Ratio)

## [1] 579

table(liver$Albumin\_and\_Globulin\_Ratio)

##   
## 0.3 0.35 0.37 0.39 0.4 0.45 0.46 0.47 0.48 0.5 0.52 0.53 0.55 0.58 0.6 0.61   
## 4 1 1 1 14 1 1 2 1 29 2 1 1 1 31 1   
## 0.62 0.64 0.67 0.68 0.69 0.7 0.71 0.74 0.75 0.76 0.78 0.8 0.87 0.88 0.89 0.9   
## 1 1 1 1 1 53 1 1 4 2 1 65 1 1 1 59   
## 0.92 0.93 0.95 0.96 0.97 1 1.02 1.03 1.06 1.09 1.1 1.11 1.12 1.16 1.18 1.2   
## 2 2 2 3 1 106 1 1 2 1 46 1 1 2 2 35   
## 1.25 1.27 1.3 1.34 1.36 1.38 1.39 1.4 1.5 1.51 1.55 1.58 1.6 1.66 1.7 1.72   
## 1 1 25 2 1 3 1 17 10 1 1 2 5 1 4 1   
## 1.8 1.85 1.9 2.5 2.8   
## 3 2 1 2 1

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.

unique(liver$Albumin\_and\_Globulin\_Ratio)

## [1] 0.90 0.74 0.89 1.00 0.40 1.30 1.10 1.20 0.80 0.60 0.87 0.70 0.92 0.55 0.50  
## [16] 1.85 0.95 1.40 1.18 0.61 1.34 1.39 1.60 1.58 1.25 0.78 0.76 1.55 0.71 0.62  
## [31] 0.67 0.75 1.16 1.50 1.66 0.96 1.38 0.52 0.47 0.93 0.48 0.58 0.69 1.27 1.12  
## [46] 1.06 0.53 1.03 0.68 1.90 1.70 1.80 0.30 0.97 0.35 1.51 0.64 0.45 1.36 0.88  
## [61] 1.09 1.11 1.72 2.80 0.46 0.39 1.02 2.50 0.37

length(unique(liver$Albumin\_and\_Globulin\_Ratio))

## [1] 69

min(liver$Albumin\_and\_Globulin\_Ratio)

## [1] 0.3

max(liver$Albumin\_and\_Globulin\_Ratio)

## [1] 2.8

Here the total observation of Albumin\_and\_Globulin\_Ratio is 579 and is a continuous variable that range lies between 0.3-2.8. Now, we will see the summary of the Albumin\_and\_Globulin\_Ratio variable, for further analysis as follows:

summary(liver$Albumin\_and\_Globulin\_Ratio)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3000 0.7000 0.9300 0.9471 1.1000 2.8000

sd(liver$Albumin\_and\_Globulin\_Ratio)

## [1] 0.3195921

var(liver$Albumin\_and\_Globulin\_Ratio)

## [1] 0.1021391

v <- c(liver$Albumin\_and\_Globulin\_Ratio)  
mode <- getMode(v)  
print(mode)

## [1] 1

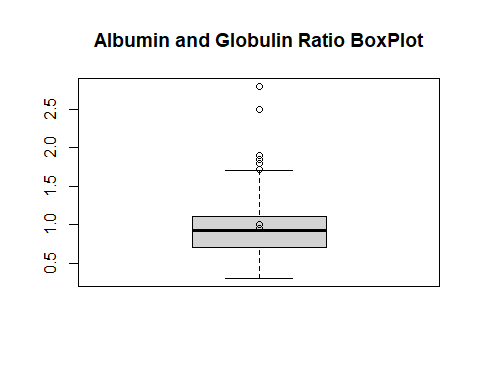
From the above summary we can observe that the Mean (0.9471), Median (0.9300) and Mode (1) are not equal so the distribution is asymmetrical. Here Mean > Median > Mode so the distribution could be positive and skewed to the right. Now we also confirm this as follows:

library(moments)  
skewness(liver$Albumin\_and\_Globulin\_Ratio)

## [1] 0.9897269

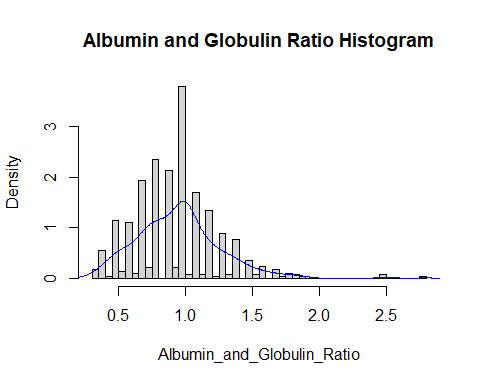
Hence, the skewness is the positive so the distribution is positively skewed was rightly observed. Also I will plot the Boxplot to comparing the distribution of data across dataset as follows:

boxplot(liver$Albumin\_and\_Globulin\_Ratio, main = "Albumin and Globulin Ratio BoxPlot")  
points(mean(liver$Albumin\_and\_Globulin\_Ratio))  
points(mode)



Box plots are graphical displays of the five number summary, So they describe the Minimum (lowest mean relative) lower whisker, Quartile 1 (It separates the lowest 25% observation to the rest of the observations), Median (Which divides the lower 50% observation from the upper 50% observations), Quartile 3 (It divides the upper 25% observation to the rest of the observations) and Maximum. In the above Boxplot diagram, there are some outliers and the upper whisker shows the highest mean relative. Now, I will plot a histogram a graphical display of data using bars of different heights to measures distribution of continuous data as follows:

hist(liver$Albumin\_and\_Globulin\_Ratio, prob = TRUE, breaks = 69, xlab = "Albumin\_and\_Globulin\_Ratio", main = "Albumin and Globulin Ratio Histogram")  
lines(density(liver$Albumin\_and\_Globulin\_Ratio), col='blue')



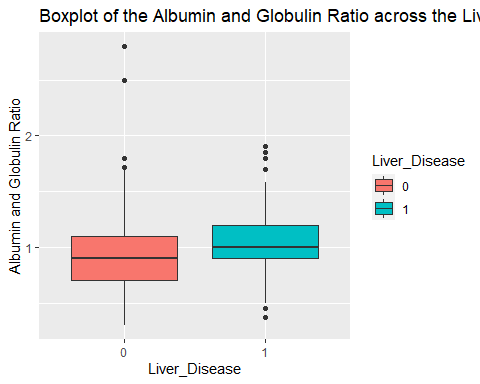
From the above histogram, we can observe that the diagram has many peaks.Using the density lines, I have approximated the histogram graph. Density provides information about the type of distribution under consideration and allows us to determine whether the attribute is distributed normally or not. From the graph above we can see how the Albumin\_and\_Globulin\_Ratio variable does not seem to fit to a Normal distribution, there is a peak of the frequency distributions around many values and also a right skewed. We have already observed a positive value of 0.9897269, so we will affirm that the distribution is right skewed. To check whether the distribution is Platykurtic or not, we will check this by following R method as follows:

kurtosis(liver$Albumin\_and\_Globulin\_Ratio)

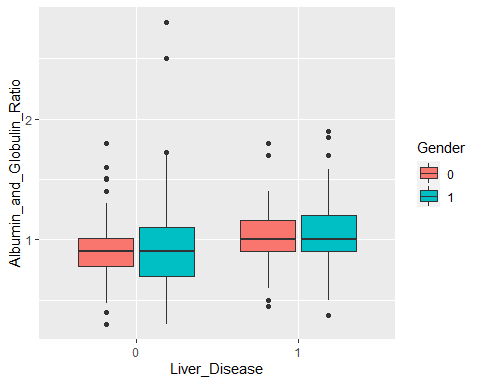
## [1] 6.243282

The distribution is also Leptokurtic, since the value is greater than 3.

ggplot(liver, aes(x = Liver\_Disease, y = Albumin\_and\_Globulin\_Ratio,  
 fill = Liver\_Disease)) +  
 geom\_boxplot() +  
 ylab("Albumin and Globulin Ratio") +  
 ggtitle("Boxplot of the Albumin and Globulin Ratio across the Liver\_Disease")



ggplot(liver, aes(Liver\_Disease, Albumin\_and\_Globulin\_Ratio)) +   
 geom\_boxplot(aes(fill = Gender))

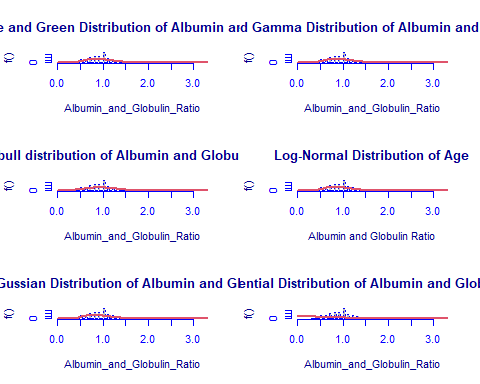


The mean for the ratio is 0.947 and is very close to the median of 0.93. The ratio has somewhat of a bell shaped curve. The mean for the ratio is higher in the Liver\_Disease = 1 response, with the range being larger in the Liver\_Disease = 0 group. There is no difference with the Liver\_Disease when you compare the genders. The ranges vary though.

### Albumin and Globulin Ratio Fit for the Data

Now we will try to fit different models to Albumin\_and\_Globulin\_Ratio distribution evaluating the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC), The lower are the indexes, the better is the fitting. I will evaluate both the single parameter distributions and mixture distributions as follows:

library(MASS)  
library(gamlss)  
  
  
par(mfrow=c(3,2))  
  
agr.BCCG <- histDist(liver$Albumin\_and\_Globulin\_Ratio, family=BCCG, nbins = 69, xlab = "Albumin\_and\_Globulin\_Ratio", main="Box-Cox Cole and Green Distribution of Albumin and Globulin Ratio")  
  
agr.GG <- histDist(liver$Albumin\_and\_Globulin\_Ratio, family=GG, nbins = 69, xlab = "Albumin\_and\_Globulin\_Ratio", main="Generalized Gamma Distribution of Albumin and Globulin Ratio")  
  
agr.WEI <- histDist(liver$Albumin\_and\_Globulin\_Ratio, family=WEI, nbins = 69, xlab = "Albumin\_and\_Globulin\_Ratio", main="Weibull distribution of Albumin and Globulin Ratio")  
  
agr.LOGNO <- histDist(liver$Albumin\_and\_Globulin\_Ratio, family=LOGNO, nbins = 69, xlab = "Albumin and Globulin Ratio", main="Log-Normal Distribution of Age")  
  
agr.IG <- histDist(liver$Albumin\_and\_Globulin\_Ratio, family=IG, nbins = 69, xlab = "Albumin\_and\_Globulin\_Ratio", main = "Inverse Gussian Distribution of Albumin and Globulin Ratio")  
  
agr.EXP<- histDist(liver$Albumin\_and\_Globulin\_Ratio, family=EXP, nbins = 69, xlab = "Albumin\_and\_Globulin\_Ratio", main = "Exponential Distribution of Albumin and Globulin Ratio")



library(Matrix)  
library(glmnet)  
df <- data.frame(Rownames = c("Box-Cox Cole and Green","Generalized Gamma", "Weibull",  
 "Log-Normal", "Inverse Gussian", "Exponential"),  
 AIC = c(AIC(agr.BCCG), AIC(agr.GG), AIC(agr.WEI), AIC(agr.LOGNO),  
 AIC(agr.IG), AIC(agr.EXP)),  
 BIC = c(agr.BCCG$sbc, agr.GG$sbc, agr.WEI$sbc, agr.LOGNO$sbc,  
 agr.IG$sbc, agr.EXP$sbc),  
 df = c(agr.BCCG$df.fit, agr.GG$df.fit, agr.WEI$df.fit,  
 agr.LOGNO$df.fit, agr.IG$df.fit, agr.EXP$df.fit),  
 LogLike = c(logLik(agr.BCCG), logLik(agr.GG), logLik(agr.WEI),  
 logLik(agr.LOGNO), logLik(agr.IG), logLik(agr.EXP)))  
df

## Rownames AIC BIC df LogLike  
## 1 Box-Cox Cole and Green 274.4812 287.5651 3 -134.2406  
## 2 Generalized Gamma 275.0929 288.1769 3 -134.5465  
## 3 Weibull 331.1582 339.8808 2 -163.5791  
## 4 Log-Normal 286.9738 295.6965 2 -141.4869  
## 5 Inverse Gussian 291.8929 300.6155 2 -143.9464  
## 6 Exponential 1097.0179 1101.3792 1 -547.5089

As we can see, the model with the highest log likelihood (-134.2406) and the lowest AIC (274.4812) and BIC (287.5651) is the Box-Cox Cole and Green Distribution. Therefore, based on the greatest likelihood technique, our data fits better in the Box-Cox Cole and Green Distribution. Now, I will also compare two models by means of the Likelihood ratio test as we performed above as follows:

library(zoo)  
library(lmtest)  
lrtest(agr.BCCG, agr.EXP)

## Likelihood ratio test  
##   
## Model 1: gamlssML(formula = liver$Albumin\_and\_Globulin\_Ratio, family = "BCCG")  
## Model 2: gamlssML(formula = liver$Albumin\_and\_Globulin\_Ratio, family = "EXP")  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -134.24   
## 2 1 -547.51 -2 826.54 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Under the null hypothesis, we compare the Box-Cox Cole and Green distribution to the Exponential distribution under the alternative hypothesis. At any level of significance, we can reject the null hypothesis. Because our distributions fit better in the Box-Cox Cole and Green distribution model, we’ll use it.

## Liver Disease

Lets analyze the features of Liver Disease variable as follows:

length(liver$Liver\_Disease)

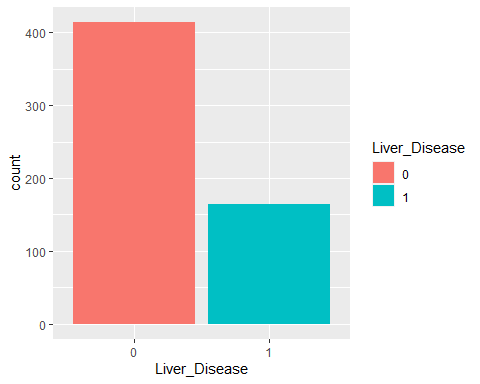
## [1] 579

table(liver$Liver\_Disease)

##   
## 0 1   
## 414 165

The table() function in R returns the absolute frequencies for each patient-specified value in the data set.Here we can see the liver$Liver\_Disease variable is a categorical variable that can take two values 0 which means no disease, 1 which means have liver disease.

library(ggplot2)  
ggplot(liver, aes(x = Liver\_Disease, fill = Liver\_Disease)) +   
geom\_bar()



As the bar plot shows our data set contains data belonging to 414 patients have no Liver Disease and 165 patients having a Liver Disease. So from both plot and table() we can see that our data is not balanced and the study includes more no disease than having disease.

# Principal Component Analysis